

INTELIGENCIA DE LOS ROBOTS



HUMMINGBIRD - 1

HIR

6.1. FUNDAMENTOS DE LA INTELIGENCIA ARTIFICIAL

En este capítulo se presentan los fundamentos de la Inteligencia Artificial (IA), conceptos relacionados, problemas, modelos, importancia, relaciones y criterios para evaluar sistemas de IA.

Inteligencia, es la aptitud de crear relaciones. Esta creación puede darse de manera puramente sensorial, como en la inteligencia animal; también puede darse de manera intelectual, como en el ser humano, que pone en juego el lenguaje y los conceptos. También se la puede conceptualizar como la habilidad para adquirir, comprender y aplicar conocimiento; o como la aptitud para recordar, pensar y razonar.

La IA es una nueva generación de tecnología informática, caracterizada no sólo por su arquitectura (*hardware*), sino también por sus capacidades. La nueva generación de tecnología informática incluye además la **manipulación simbólica**, con el objetivo de emular el comportamiento inteligente; y, la **computación en paralelo**, para tratar de conseguir resultados prácticamente en tiempo real. La capacidad predominante de la nueva generación, también conocida como la **Quinta Generación**, es la habilidad de emular (y tal vez en algunos casos superar) ciertas funciones inteligentes del ser humano.

Por ejemplo:

Aprendizaje:

- Captación automática de conocimientos.

Razonamiento:

- Sistemas basados en conocimientos.
- Bases de datos inteligentes.
- Prueba de teoremas y juegos.

Percepción:

- Comprensión de lenguaje natural.
- Interpretación de escenas visuales (Visión por computadora).

Locomoción y Manipulación:

- Realizar procesos mecánicos y tareas manuales (Robótica).

Creación:

- Generación, verificación, depuración y optimización automática de programas.

Algunas de las tareas que estos sistemas realizan en el campo de la IA son:

Tareas generales

- Percepción: Visión, Fonemas.
- Lenguaje Natural: Comprensión, generación y traducción.
- Razonamiento de sentido común.
- Control de robots.

Tareas formales

- Juegos: Ajedrez, Damas.
- Matemáticas: Geometría, Lógica, Cálculo Integral.

Tareas expertas

- Ingeniería: Diseño, Localización de fallas, Planeamiento.
- Análisis Científico.
- Diagnóstico Médico.
- Análisis Financiero.

6.2. DEFINICIONES DE INTELIGENCIA ARTIFICIAL

Primero veamos algunas definiciones generales que diversos autores proponen desde las diferentes perspectivas que, cada autor cree, encierra el campo. A continuación se transcriben algunas de ellas:

"La IA es una rama de la ciencia de computación que comprende el estudio y creación de sistemas computarizados que manifiestan cierta forma de inteligencia: sistemas que aprenden nuevos conceptos y tareas, sistemas que pueden razonar y derivar conclusiones útiles acerca del mundo que nos rodea, sistemas que pueden comprender un lenguaje natural o percibir y comprender una escena visual, y sistemas que realizan otro tipo de actividades que requieren de inteligencia humana¹."

La IA es una ciencia que trata de la comprensión de la inteligencia y del diseño de máquinas inteligentes, es decir, el estudio y la simulación de las actividades intelectuales del hombre (manipulación, razonamiento, percepción, aprendizaje, creación).

La IA es el estudio de las computaciones que permiten percibir, razonar y actuar.

¹ K.S. F.U, R.C. GONZALES (1988) "Robótica". Capitulo 11, Pagina 15

La IA es un campo de estudio que busca explicar y emular el comportamiento inteligente en términos de procesos computacionales.

La IA estudia las representaciones y procedimientos que automáticamente resuelven problemas usualmente resueltos por humanos²

A pesar de la diversidad de conceptos propuestos para la IA, en general todos coinciden en que la IA trata de alcanzar inteligencia a través de la computación. Toda computación, requiere de una **representación** de cierta entidad y de un proceso para su **manipulación**.

Desde el punto de vista de los objetivos, la IA puede considerarse en parte como ingeniería y en parte como ciencia:

Como **ingeniería**, el objetivo de la IA es resolver problemas reales, actuando como un conjunto de ideas acerca de cómo representar y utilizar el conocimiento, y de cómo desarrollar sistemas informáticos.

Como **ciencia**, el objetivo de la IA es buscar la explicación de diversas clases de inteligencia, a través de la representación del conocimiento y de la aplicación que se da a éste en los sistemas informáticos desarrollados.

Para usar la IA se requiere una comprensión básica de la forma en que se puede representar el conocimiento y de los métodos que pueden utilizar o manipular ese conocimiento.

6.3. IMPORTANCIA DE LA INTELIGENCIA ARTIFICIAL

Conforme el mundo se vuelve más complejo, debemos usar nuestros recursos materiales y humanos con más eficiencia, y para lograrlo, se necesita la ayuda que nos ofrecen los computadores.

La IA está todavía en vías de desarrollo. La IA se está dividiendo y encontrando otros campos como: lógica, redes neuronales, programación orientada a objetos, lenguajes formales, robótica, etc. Esto explica por qué el estudio de IA no está confinado a la matemática, ciencias de la computación, ingeniería (particularmente la electrónica y la

² www.IA.com

mecánica), o a la ciencia cognoscitiva, sino que cada una de estas disciplinas es un potencial contribuyente. La robótica es considerada como un campo interdisciplinario que combina conceptos y técnicas de IA, con ingeniería óptica, electrónica y mecánica

6.4. PROBLEMAS Y TECNICAS EN INTELIGENCIA ARTIFICIAL

Desde el punto de vista de ingeniería, la mayor parte del trabajo requerido para construir sistemas de IA, está basado en el desarrollo de adecuadas *representaciones de conocimiento* y sus correspondientes *estrategias de manipulación*. **No se puede manipular conocimiento a menos que esté adecuadamente representado.**

6.4.1. CONOCIMIENTO

Puede ser definido como el conjunto de hechos y principios acumulados por la humanidad, o el acto, hecho o estado de conocer. Es la familiaridad con el lenguaje, conceptos, procedimientos, reglas, ideas, abstracciones, lugares, costumbres y asociaciones, unida a la habilidad de utilizar estas nociones en forma efectiva para modelar diferentes aspectos del universo que nos rodea.

Los conceptos de conocimiento e inteligencia están íntimamente ligados. La inteligencia requiere de la posesión y acceso al conocimiento. Conocimiento no debe ser confundido con datos o información. El conocimiento incluye y requiere del uso de datos e información. Además, combina relaciones, dependencias, y la noción del saber con datos e información.

A veces también es útil o más aún necesario distinguir entre conocimiento y otros términos como *creencia* e *hipótesis*.

Esencialmente, se define como **creencia** a toda expresión que tiene significado, que es coherente y puede ser representada. En consecuencia una creencia puede ser verdadera o falsa. Hipótesis se define como una creencia justificada que no se conoce sea verdadera. Una **hipótesis** es una creencia que está respaldada por cierta evidencia, pero todavía puede resultar falsa. Finalmente, se puede también decir que conocimiento es una creencia justificada como verdadera.

El conocimiento puede ser de tipo *procedimental*, *declarativo* o *heurístico*. **Conocimiento procedimental** es aquel conocimiento compilado que se refiere a la forma de realizar

una cierta tarea (*el saber como hacerlo*). Por ejemplo, los pasos necesarios para resolver una ecuación algebraica son expresados como conocimiento procedimental.

Por otro lado, el **conocimiento declarativo** es conocimiento pasivo, expresado como sentencias acerca de los hechos del mundo que nos rodea (*el saber que hacer*). La información personal en una base de datos es un típico ejemplo de conocimiento declarativo. Tales tipos de datos son piezas explícitas de conocimiento independiente.

El **conocimiento heurístico** es un tipo especial de conocimiento usado por los humanos para resolver problemas complejos. El adjetivo *heurístico* significa “*medio para descubrir*”. Está relacionado con la palabra griega *heuriskein* que significa descubrir, encontrar. Se entiende por *heurístico* a un criterio, estrategia, método o truco utilizado para simplificar la solución de problemas. El conocimiento heurístico usualmente se lo adquiere a través de mucha experiencia.

6.5. REPRESENTACION DEL CONOCIMIENTO

El conocimiento puede ser representado como imágenes mentales en nuestros pensamientos, como palabras habladas o escritas en algún lenguaje, en forma gráfica o en imágenes, como cadenas de caracteres o colecciones de señales eléctricas o magnéticas dentro de un computador.

En organismos biológicos se estima que el conocimiento es almacenado como estructuras complejas de neuronas interconectadas.

En las computadoras, el conocimiento también se almacena como estructuras simbólicas, pero en forma de estados eléctricos y magnéticos.

En forma natural, el ser humano representa el conocimiento simbólicamente: imágenes, lenguaje hablado y lenguaje escrito. Adicionalmente, ha desarrollado otros sistemas de representación del conocimiento: literal, numérico, estadístico, estocástico, lógico.

Los modelos sistemas de representación del conocimiento implantados en un computador, se aproximan mucho a los modelos elaborados por la psicología cognoscitiva para el cerebro humano. Tales como:

Lógica Simbólica Formal:

- Lógica proposicional.
- Lógica de predicados.
- Reglas de producción.

Formas Estructuradas:

- Redes asociativas.
- Representación mediante Plantillas.
- Representación orientada a objetos.

6.5.1. LOGICA PROPOSICIONAL

La lógica proposicional es la más antigua y simple de las formas de lógica. Utilizando una representación primitiva del lenguaje, permite representar y manipular aserciones sobre el mundo que nos rodea. La lógica proposicional permite el razonamiento, a través de un mecanismo que primero evalúa sentencias simples y luego sentencias complejas, formadas mediante el uso de conectivos proposicionales, Y (AND) (\wedge), O (OR) (\vee). Este mecanismo determina la veracidad de una sentencia compleja, analizando los valores de veracidad asignados a las sentencias simples que la conforman.

Una proposición es una sentencia simple que tiene un valor asociado ya sea de verdadero (V), o falso (F). Por ejemplo:

Hoy es Martes
Ayer llovió
Hace frío

La lógica proposicional, permite la asignación de un valor verdadero o falso para la sentencia completa, no tiene facilidad par analizar las palabras individuales que componen la sentencia. Por este motivo, la representación de las sentencias del ejemplo, como proposiciones, sería:

hoy_es_Martes
ayer_llovió
hace_frío

Las proposiciones pueden combinarse para expresar conceptos más complejos. Por ejemplo:

hoy_es_Martes y hace_frío.

A la proposición anterior dada como ejemplo, se la denomina fórmula bien formada. Una fórmula bien formada puede ser una proposición simple o compuesta que tiene sentido completo y cuyo valor de veracidad, puede ser determinado. La lógica proposicional proporciona un mecanismo para asignar valores de veracidad a la proposición compuesta, basado en los valores de veracidad de las proposiciones simples y en la naturaleza de los conectores lógicos involucrados.

NOMBRE	CONECTOR	SIMBOLO
Conjunción	AND	\wedge
Disyunción	OR	\vee
Negación	NOT	\sim
Implicación	If-Then	\Rightarrow
Equivalencia	Igual	$=$

Tabla 6.1 Conectores básicos de la lógica proposicional

p	q	Disyunción $p \vee q$	Conjunción $p \wedge q$	Negación $\sim p$	Implicación $p \Rightarrow q$	Equivalencia $P = q$
V	V	V	V	F	V	V
V	F	V	F	F	F	F
F	V	V	F	V	V	F
F	F	F	F	V	V	V

Tabla 6.2 Tablas de verdad para operadores lógicos

El conector de implicación, puede ser considerado como un condicional expresado de la siguiente forma:

Si $A \Rightarrow B$ va a ser verdadero, entonces toda vez que A sea verdadero, B debe ser siempre verdadero.

Existen varias equivalencias en lógica proposicional, similares a las del álgebra Booleana.

DENOMINACION	REPRESENTACION LOGICA
Leyes Equipotenciales	$A \Rightarrow B = \sim A \vee B$ $A \wedge \sim A = F$ $A \vee \sim A = V$
Leyes Conmutativas	$A \wedge B = B \wedge A$ $A \vee B = B \vee A$
Leyes Distributivas	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
Leyes Asociativas	$A \wedge (B \wedge C) = (A \wedge B) \wedge C$ $A \vee (B \vee C) = (A \vee B) \vee C$
Leyes Absortivas	$A \wedge (A \vee B) = A$ $A \vee (A \wedge B) = A$
Leyes de DeMorgan	$\sim(A \wedge B) = \sim A \vee \sim B$ $\sim(A \vee B) = \sim A \wedge \sim B$

Tabla 6.3 Equivalencias en lógica proposicional

6.5.2. LOGICA DE PREDICADOS

La lógica de predicados está basada en la idea de las sentencias realmente expresan relaciones entre objetos, así como también cualidades y atributos de tales objetos. Los objetos pueden ser personas, objetos físicos, o conceptos. Tales *cualidades, relaciones* o *atributos*, se denominan **predicados**. Los *objetos* se conocen como **argumentos o términos** del predicado.

Al igual que las proposiciones, los predicados tienen un valor de veracidad, pero a diferencia de las preposiciones, su valor de veracidad, depende de sus términos. Es decir, un predicado puede ser verdadero para un conjunto de términos, pero falso para otro. Por ejemplo, el siguiente predicado es verdadero:

color (yerba, verde)

El mismo predicado, pero con diferentes argumentos, puede no ser verdadero:

color (yerba, azul) o color (cielo, verde)

Los predicados también pueden ser utilizados para asignar una cualidad abstracta a sus términos, o para representar acciones o relaciones de acción entre dos objetos. Por ejemplo:

ama(roberto, vanessa) Clima(martes, lluvioso) ave(gaviota)

Al construir los predicados se asume que su veracidad está basada en su relación con el mundo real. Naturalmente que los predicados que definimos estén de acuerdo con el mundo que conocemos, pero no es absolutamente necesario que así lo hagamos. En lógica de predicados el establecer como verdadero un predicado es suficiente para que así sea considerado. Demos el siguiente ejemplo, que indica que Ecuador está en Europa:

parte_de(ecuador, europa)

Obviamente, esto no es verdadero en el mundo real, pero la lógica de predicados no tiene razón de saber geografía y si el predicado es dado como verdadero, entonces es considerado como lógicamente verdadero. Tales predicados, establecidos y asumidos como lógicamente verdaderos se denominan *axiomas*, y no requieren de justificación para establecer su verdad.

La lógica de predicados, se ocupa únicamente de métodos de argumentación sólidos. Tales argumentaciones se denominan **Reglas de Inferencia**. Si se da un conjunto de axiomas que son aceptados como verdaderos, las reglas de inferencia garantizan que sólo serán derivadas consecuencias verdaderas.

Tanto los conectivos lógicos, como los operadores dados anteriormente para la lógica proposicional, son igualmente válidos en lógica de predicados. De hecho, la lógica proposicional es un subconjunto de la lógica de predicados.

Cada uno de los argumentos en los ejemplos de predicados dados anteriormente, representan a un objeto específico. Tales argumentos se denominan *constantes*. Sin embargo, en la lógica de predicados se pueden tener argumentos que en determinado momento pueden ser desconocidos. Estos son los argumentos tipo *variable*.

En el ejemplo: *color (yerba, X)*, la variable *X*, puede tomar el valor de *verde*, haciendo que el predicado sea verdadero; o puede tomar el valor de *azul*, dando lugar a que el predicado sea falso.

Las variables, también pueden ser cuantificadas. Los cuantificadores que típicamente se utilizan en lógica de predicados son:

$\forall \cdot X \dots$	$\exists X \dots$
El cuantificador universal \forall , indica que la fórmula bien formada, dentro de su alcance, es verdadera para todos los valores posibles de la variable que es cuantificada. Por ejemplo:	El cuantificador existencial \exists , indica que la fórmula bien formada, dentro de su alcance, es verdadera para algún valor o valores dentro del dominio.
Establece que "para todo X, es verdad que . . ."	Establece que "existe un X, tal que . . ."

Tabla 6.4 Cuantificadores y su significado

A continuación se dan algunos ejemplos de predicados cuantificados:

- $\forall \cdot X, [niño(X) \Rightarrow le_gusta(X, helados)].$
- $\cdot \forall Y, [mamífero(Y) \Rightarrow nace(Y, vivo)].$
- $\cdot \exists Z, [cartero(Z) \wedge mordió(boby, Z)].$

El cuantificador universal es más simple ya que se asume a todas las variables como universalmente cuantificadas.

El cuantificador existencial es más difícil de reemplazar. El cuantificador existencial garantiza la existencia de uno o más valores particulares (*instancias*) de la variable cuantificada, que hace a la cláusula verdadera. Si se asume que existe una función capaz de determinar los valores de la variable que hace la cláusula verdadera, entonces simplemente se remueve el cuantificador existencial y se reemplaza las variables por la función que retorna dichos valores.

Se debe prestar especial atención a un conjunto de términos que hacen que el comportamiento del valor que adquiera una proposición sea determinante. Tal como:

a) UNIFICACION

Cuando se tienen sentencias compuestas por predicados y conectivos lógicos, se debe evaluar la veracidad de cada uno de sus componentes para determinar si toda la sentencia es verdadera o falsa. Para ello, se busca en el conjunto de axiomas la forma de establecer la veracidad de los predicados componentes. Un predicado componente se dice que es verdadero si se identifica con un axioma de la base de información.

b) INFERENCIA Y RAZONAMIENTO

Inferir es concluir o decidir a partir de algo conocido o asumido. Por otro lado, **razonar** es pensar coherente y lógicamente a partir de hechos conocidos o asumidos.

El proceso de razonamiento, involucra la realización de inferencias, a partir de hechos conocidos. Realizar inferencias significa derivar nuevos hechos a partir de un conjunto de hechos conocidos como verdaderos. La lógica de predicados proporciona un grupo de reglas sólidas, con las cuales se pueden realizar inferencias.

Las **principales Reglas de Inferencia** son:

- **Modus ponens.**- Es la más importante, en los sistemas basados en conocimiento. Establece que:

Si las sentencias p y $(p \rightarrow q)$ se conocen que son verdaderas, entonces se puede inferir que q también es verdadera.

- **Modus tolens.**- Esta regla establece que:

Si la sentencia $(p \rightarrow q)$ es verdadera y q es falsa, entonces se puede inferir que p también es falsa.

Utiliza refutación para comprobar una determinada sentencia. La refutación intenta crear una contradicción con la negación de la sentencia original, demostrando, por lo tanto, que la sentencia original es verdadera. La resolución es una técnica poderosa para probar teoremas en lógica y constituye la técnica básica de inferencia.

6.5.3. REGLAS DE PRODUCCION

Los sistemas basados en reglas son los más comúnmente utilizados. Su simplicidad y similitud con el razonamiento humano, han contribuido para su popularidad en diferentes dominios. Las reglas son un importante paradigma de representación del conocimiento.

Las reglas representan el conocimiento utilizando un formato **SI-ENTONCES (IF-THEN)**, es decir tienen 2 partes:

La parte **SI (IF)**, es el antecedente, premisa, condición o situación, y la parte **ENTONCES (THEN)**, es el consecuente, conclusión, acción o respuesta.

Las reglas pueden ser utilizadas para expresar un amplio rango de asociaciones, por ejemplo:

SI *está manejando un vehículo Y se aproxima una ambulancia,* **ENTONCES** *baje la velocidad Y hágase a un lado para permitir el paso de la ambulancia.*

SI *su temperatura corporal es de 39 °C,* **ENTONCES** *tiene fiebre.*

SI *el drenaje del lavabo está tapado Y la llave de agua está abierta,* **ENTONCES** *se puede inundar el piso.*

a) INFERENCIA BASADA EN REGLAS

Una declaración de que algo es verdadero o es un hecho conocido, es una **afirmación**. El conjunto de afirmaciones se conoce a menudo con el nombre de **base de afirmaciones**. De igual forma, al conjunto de reglas se lo denomina **base de reglas**.

Un sistema basado en reglas utiliza el *modus ponens* para manipular las afirmaciones y las reglas durante el proceso de inferencia. Mediante técnicas de búsqueda y procesos de unificación, los sistemas basados en reglas automatizan sus métodos de razonamiento y proporcionan una progresión lógica desde los datos iniciales, hasta las conclusiones deseadas. Esta progresión hace que se vayan conociendo nuevos hechos o descubriendo nuevas afirmaciones, a medida que va guiando hacia la solución del problema.

b) EL PROCESO DE RAZONAMIENTO

El proceso de razonamiento en un sistema basado en reglas es una progresión desde un conjunto inicial de afirmaciones y reglas hacia una solución, respuesta o conclusión. Como se llega a obtener el resultado, sin embargo, puede variar significativamente:

Se puede partir considerando todos los datos conocidos y luego ir progresivamente avanzando hacia la solución. Este proceso se lo denomina *guiado por los datos* o de **encadenamiento progresivo**.

Se puede seleccionar una posible solución y tratar de probar su validez buscando evidencia que la apoye. Este proceso se denomina *guiado por el objetivo* o de **encadenamiento regresivo**.

c) RAZONAMIENTO PROGRESIVO

Se empieza a partir de un conjunto de datos colectados a través de observación y se evoluciona hacia una conclusión. Se chequea cada una de las reglas para ver si los datos observados satisfacen las premisas de alguna de las reglas. Si una regla es satisfecha, es ejecutada derivando nuevos hechos que pueden ser utilizados por otras reglas para derivar hechos adicionales. Este proceso de chequear reglas para ver si pueden ser satisfechas se denomina *interpretación de reglas*.

Un conjunto de aplicaciones adecuadas al razonamiento progresivo incluye supervisión y diagnóstico en sistemas de control de procesos en tiempo real, donde los datos están continuamente siendo adquiridos, modificados y actualizados.

Otro conjunto de aplicaciones adecuadas para el razonamiento progresivo está formado por: diseño, planeamiento y calendarización, donde ocurre la síntesis de nuevos hechos basados en las conclusiones de las reglas.

d) RAZONAMIENTO REGRESIVO

El razonamiento regresivo, difiere significativamente del mecanismo de razonamiento progresivo. Si bien es cierto, ambos procesos involucran el examen y aplicación de reglas, el razonamiento regresivo empieza con la conclusión deseada y decide si los hechos que existen pueden dar lugar a la obtención de un valor para esta conclusión. El razonamiento regresivo sigue un proceso muy similar a la búsqueda primero en profundidad.

El sistema empieza con un conjunto de hechos conocidos que típicamente está vacío. Se proporciona una lista ordenada de conclusiones, para las cuales el sistema trata de derivar valores. El proceso de razonamiento regresivo utiliza esta lista de objetivos para coordinar su búsqueda a través de las reglas de la base de conocimientos.

➤ Algoritmo de Búsqueda

- 1.** Conformar una pila inicialmente compuesta por todos los objetivos prioritarios definidos en el sistema.
- 2.** Considerar el primer objetivo de la pila. Determinar todas las reglas capaces de satisfacer este objetivo, es decir aquellas que mencionen al objetivo en su conclusión.

3. Para cada una de estas reglas examinar en turno sus antecedentes:

- Si todos los antecedentes de la regla son satisfechos (esto es, cada parámetro de la premisa tiene su valor especificado dentro de la base de datos), entonces ejecutar esta regla para derivar sus conclusiones. Debido a que se ha asignado un valor al objetivo actual, removerlo de la pila y retornar al paso (2).

Si alguna premisa de la regla no puede ser satisfecha, buscar reglas que permitan derivar el valor especificado para el parámetro utilizado en esta premisa.

- Si en el paso (b) no se puede encontrar una regla para derivar el valor especificado para el parámetro actual, entonces preguntar al usuario por dicho valor y añadirlo a la base de datos. Si este valor satisface la premisa actual entonces continuar con la siguiente premisa de la regla. Si la premisa no es satisfecha, considerar la siguiente regla.
- Una excelente aplicación para el razonamiento regresivo es el diagnóstico, donde el usuario dialoga directamente con el sistema basado en conocimiento y proporciona los datos.

e) REDES DE INFERENCIA

Una red de inferencia puede ser representada como un gráfico, en el que los nodos representan parámetros que son los hechos obtenidos como datos o derivados de otros datos. Cada parámetro es una declaración acerca de algún aspecto del problema y puede servir como un antecedente o consecuente de una regla. Estas declaraciones pueden copar un rango que va desde la conclusión final de un sistema, hasta hechos simples, observados o derivados. Cada uno de estos parámetros puede tener uno o más valores asociados, donde cada valor tiene una medida correspondiente de incertidumbre que representa cuán creíble es el valor particular de un parámetro.

Las reglas en el sistema están representadas dentro del gráfico por las interconexiones entre los varios nodos. Este conocimiento es utilizado por el proceso de inferencia para propagar resultados a través de la red.

Las redes de inferencia son muy útiles para dominios donde el número de diferentes soluciones alternativas es limitado. Por ejemplo, la clasificación de elementos en las ciencias naturales y problemas de diagnóstico. Una red de inferencia es fácil de

implementar, pero es menos poderosa ya que se debe conocer de antemano todas las relaciones entre reglas y hechos.

6.5.4. REDES ASOCIATIVAS

Las redes asociativas se caracterizan por representar el conocimiento en forma gráfica. Agrupan una porción de conocimiento en dos partes: objetos y relaciones entre objetos. Los objetos se denominan también nodos (elementos del conocimiento) y las relaciones entre nodos se denominan enlaces o arcos. Cada nodo y cada enlace en una red semántica, deben estar asociados con objetos descriptivos.

Son muy apropiadas para representar conocimiento de naturaleza jerárquica. Su concepción se basa en la asociación de conocimientos que realiza la memoria humana. Las principales aplicaciones son: comprensión de lenguaje natural, bases de datos deductivas, visión por computadora, sistemas de aprendizaje.

6.5.5. MEDIANTE PLANTILLAS

Una *plantilla* representa un objeto o situación describiendo la colección de atributos que posee. Están formadas por un nombre y por una serie de *campos* de información. Cada campo puede contener uno o más *enlaces*. Cada enlace tiene un valor asociado. Varios enlaces pueden ser definidos para cada campo.

Además los enlaces pueden ser procedimientos que residen en la base de datos y están aguardando para ser utilizados cuando se los necesite. Entre los más comunes se pueden mencionar:

- *Si-Necesitado*.- Procedimiento(s) para determinar el valor actual de un campo.
- *Si-Agregado*.- Procedimiento(s) a ejecutarse cuando un valor es especificado para un campo.
- *Si-Modificado*.- Procedimiento(s) a ejecutarse si el valor de un campo es cambiado.

Estos procedimientos representan un concepto poderoso en las plantillas, esto es, la habilidad de combinar conocimiento procedimental dentro de la estructura de conocimiento declarativo de la plantilla. Esto sugiere que una plantilla puede ser un medio poderoso de representación del conocimiento, especialmente si se la incorpora en una red de plantillas.

Se pueden establecer ciertas similitudes entre un sistema basado en plantillas y un sistema de bases de datos. Aparentemente los dos representan "datos" (a través de los campos de una plantilla y de los campos de una tabla de datos), sin embargo las plantillas representan en realidad conocimiento, mientras que las bases de datos representan sólo datos.

6.5.6. MEDIANTE OBJETOS

Los objetos, son similares a las plantillas. Ambos sirven para agrupar conocimiento asociado, soportan herencia, abstracción y el concepto de procedimientos agregados.

Un **objeto** es definido como una colección de información representando una entidad del mundo real y una descripción de cómo debe ser manipulada esta información, esto es los **métodos**. Es decir, un objeto tiene un nombre, una caracterización de clase, varios atributos distintivos y un conjunto de operaciones. La relación entre los objetos viene definida por los **mensajes**. Cuando un objeto recibe un mensaje válido, responde con una acción apropiada, retornando un resultado.

<i>NOMBRE OBJETO</i>	<i>Limpiador Izquierdo</i>
<i>UN-TIPO-DE</i>	<i>Limpiador</i>
<i>ATRIBUTOS</i>	<i>Hecho de metal y caucho Longitud: 14 pulgadas Color: negro y plateado Localización: inferior izquierda Función: remover humedad de parabrisa</i>
<i>OPERACIONES</i>	<i>Activado: se mueve en arco sobre el parabrisa repetidamente de izquierda a derecha Desactivado: se mueve a posición de reposo</i>

Tabla 6.5 Descripción tipo objeto para un limpia parabrisas.

6.6. MANIPULACION DEL CONOCIMIENTO

Existen tres paradigmas que los investigadores han utilizado tradicionalmente para la resolución de problemas de IA:

- **Programación Heurística.**- Está basado en el modelo de comportamiento humano y su estilo para resolver problemas complejos. Existen diversos tipos de programas

que incluyen algoritmos heurísticos. Varios de ellos son capaces de aprender de su experiencia.

- **Redes Neuronales Artificiales.**- Es una representación abstraída del modelo neuronal del cerebro humano. Las redes están formadas por un gran número de elementos simples y por sus interconexiones. Una red neuronal artificial puede ser simulada o ser real. Al elemento procesador de la red, se lo denomina neurona artificial.
- **Evolución Artificial.**- Su modelo está basado en el proceso genético de evolución natural, propuesto por Charles Darwin. Se utilizan sistemas simulados en computador que evolucionan mediante operaciones de reproducción, mutación y cruce (Algoritmos Genéticos).

Cada paradigma comprende una colección de métodos, configuraciones y técnicas desarrolladas para manipular el conocimiento. En general, una técnica de IA está caracterizada por incluir los siguientes componentes:

- **Procesos de Búsqueda.**- Proporciona una forma de resolver problemas para los cuales no hay un método más directo, así como también se constituye en un marco de trabajo dentro del cual cualquier técnica directa puede ser incorporada.

Uso del Conocimiento.- Proporciona una forma de resolver problemas explotando las estructuras de los objetos involucrados.

- **Abstracción.**- Proporciona una forma de separar rasgos importantes y variaciones, de los tantos que no tienen importancia.

La manipulación del conocimiento involucra además la selección de objetos, entidades y rasgos que son claves para las representaciones. Al nivel más bajo de representación del conocimiento, estas entidades pueden consistir de simples datos de entrada, por ejemplo: grabaciones cuantizadas y digitalizadas de audio, datos de una imagen, valores captados por un sensor, información de interruptores o hechos elementales. Estos bloques constructivos de la representación se los denomina comúnmente *primitivas*. La representación escogida para las primitivas, puede determinar las capacidades del sistema, el éxito, la corrección y sus posibilidades de expansión.

Es muy difícil determinar si una máquina o un sistema tienen inteligencia o puede pensar. Una prueba para reconocer *comportamientos inteligentes*, es la siguiente:

Si la ejecución de un sistema de IA puede convencernos de que su comportamiento es el que tendría un humano, entonces el sistema es verdaderamente inteligente.

Debe verificarse la eficacia en el desempeño, eficiencia operativa, facilidad de uso, flexibilidad, portabilidad, etc.

6.6.1. PROCESOS HEURISTICOS DE BUSQUEDA

Para resolver muchos problemas difíciles, es necesario muchas veces llegar a un compromiso de los requerimientos de movilidad y sistematicidad y construir una estructura de control que no necesariamente garantiza el encontrar la mejor respuesta, sino que casi siempre encuentra una buena respuesta. Las consideraciones que sirven de soporte a un proceso de búsqueda heurística, son:

- Rara vez se requiere, en realidad, una solución óptima. Una buena aproximación, normalmente, sirve muy bien.
- A pesar que una aproximación heurística no puede resultar muy buena en el peor de los casos, raras veces aparecen los peores casos en la práctica.
- El tratar de comprender por qué un heurístico funciona o por qué no funciona, a menudo conduce a una mejor comprensión del problema.

Existen varias técnicas heurísticas buenas, de propósito general, que son útiles para una diversidad de problemas. Adicionalmente, es posible construir heurísticos especiales que exploten conocimiento específico en cada dominio, para resolver problemas particulares.

Ascenso a Colina (Hill Climbing)

Es una variante del algoritmo de búsqueda de generación y prueba. Del procedimiento de prueba existe una realimentación que ayuda al generador a decidirse por cual dirección debe moverse en el espacio de búsqueda. En estos procesos se abandona la búsqueda si no existe un estado alternativo razonable al que se pueda mover.

Los algoritmos de **ascenso a colina** son típicamente **locales**, ya que deciden qué hacer, mirando únicamente a las consecuencias inmediatas de sus opciones. Puede que nunca lleguen a encontrar una solución, si son atrapados en estados que no son el objetivo, desde donde no se puede hallar mejores estados, por ejemplo:

Un máximo local, que es un estado mejor que sus vecinos pero no es mejor que otros que están algo más alejados.

Una meseta, es un espacio de búsqueda en el que todo un conjunto de estados vecinos tienen igual valor.

Un risco, que es un tipo especial de máximo local, imposible de atravesar con movimientos simples.

Hay algunas formas que pueden ayudar a solventar estos problemas, aunque no existe garantía: Para evitar máximos locales, regresar a un estado anterior y explorar en una dirección diferente.

Para casos de mesetas, dar un salto grande en alguna dirección y tratar de encontrar una nueva sección del espacio de estado.

Para los riscos, aplicar dos o más reglas, antes de realizar una prueba del nuevo estado, esto equivale a moverse en varias direcciones a la vez.

Los algoritmos de ascenso a colina, a pesar de explorar sólo un paso adelante, al examinar el nuevo estado pueden incluir una cierta cantidad de información global.

Primero el Mejor (Best-First)

Este algoritmo, combina las ventajas de los algoritmos primero en profundidad y primero en amplitud. Sigue un sendero a la vez, pero puede cambiarse a otro sendero que parece más prometedor que el que está siguiendo.

En este sentido, puede considerarse que es un algoritmo que realiza su proceso de búsqueda en un **grafo** de tipo **O**, ya que todos sus ramales representan una alternativa de solución. Para su operación, el algoritmo necesita dos listas de nodos y una función heurística que estime los méritos de cada nodo que se genere:

Nodos Abiertos - Es una variable que contiene los nodos que han sido generados. La función heurística ha sido aplicada a ellos, pero todavía no han sido examinados, es decir no se han generado sus sucesores. Abiertos puede considerarse como una cola de prioridades en la que los elementos con mayor prioridad son los que tienen los valores más prometedores, dados por la función heurística.

Nodos Cerrados - Es una variable que contiene los nodos que han sido examinados. Es necesario tener esta información, para que la búsqueda sea en un grafo y no en un árbol.

Función Heurística - Permite que el algoritmo busque primero por senderos que son o parecen más prometedores.

6.6.2. REDES NEURONALES

➤ INTRODUCCION

Las redes neuronales son otra forma de emular otra de las características propias de los humanos: la capacidad de memorizar y asociar hechos. Si examinamos con atención aquellos problemas que no pueden expresarse a través de un algoritmo nos daremos cuenta de que todos ellos tienen una característica común: la experiencia. El hombre es capaz de resolver estas situaciones acudiendo a la experiencia acumulada. Así, parece claro que una forma de aproximarse al problema consista en la construcción de sistemas que sean capaces de reproducir esta característica humana. En definitiva, las redes neuronales no son más que un modelo artificial y simplificado del cerebro humano, que es el ejemplo más perfecto del que disponemos de sistema que es capaz de adquirir conocimiento a través de la experiencia. Una red neuronal es *un nuevo sistema para el tratamiento de la información cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano, la neurona.*

La programación de robots suele hacerse en relación con las coordenadas cartesianas del espacio de trabajo del robot, por lo tanto caerá en el controlador la tarea de traducir dichas coordenadas en variables articulares o motoras que gobernarán los movimientos del robot. Por lo tanto, el control de robots depende de la disponibilidad de funciones que permitan pasar del espacio físico al espacio de variables articulares o motoras.

Una función básica es la denominada cinemática inversa, que hace corresponder una posición y orientación del elemento terminal a cada vector de valores de las variables articulares. Otra función es la llamada dinámica inversa que relaciona la trayectoria del elemento terminal con las fuerzas y pares ejercidos en las distintas articulaciones.

También existe el llamado mapa sensoriomotor , que relaciona patrones sensoriales (recogidos por algún sensor en el campo de trabajo) con las ordenes motoras necesarias, un caso particular es la evitación de obstáculos por el robot móvil.

Así, las principales *características* que diferencian a las *redes neuronales* de otras tecnologías de IA son:

- Su capacidad de **aprendizaje** a partir de la experiencia (*entrenamiento*). Normalmente, para la elaboración de un *programa informático* es necesario un *estudio* detallado de la tarea a realizar para después *codificarla* en un lenguaje de programación. Pero, las *redes neuronales* pueden ser *entrenadas* para realizar una determinada tarea sin necesidad de un estudiar esta a fondo ni programarla usando un lenguaje de programación. Además; las redes neuronales pueden volver a entrenarse para ajustarse a nuevas necesidades de la tarea que realizan, sin tenerse que reescribir o revisar el código (cosa frecuente en programas tradicionales).
- Su **velocidad** de respuesta una vez concluido el entrenamiento. Se comportan también en este caso de manera similar a como lo hace el cerebro: los seres humanos no necesitamos pensar mucho para identificar un objeto, una palabra,... una vez hemos aprendido a hacerlo.
- Su **robustez**, en el sentido de que el conocimiento adquirido se encuentra repartido por toda la red, de forma que si se lesiona una parte se continúan generando cierto número de respuestas correctas (en este caso también hay cierta analogía con los cerebros parcialmente dañados).

Las características de las redes neuronales hacen que sus posibles aplicaciones sean muy amplias. Algunas de las aplicaciones más destacadas son:

- Reconocimiento de patrones de clasificación:
 - Reconocimiento de voz, de caracteres manuscritos.
 - Análisis y reconocimiento de imágenes, formas.
 - Diagnóstico clínico.
- Análisis de series temporales y predicción:
 - Modelos meteorológicos.
 - Predicción del comportamiento.
 - Predicción de series temporales.
- En la Robótica

A la hora de implementar una red neuronal como parte de un programa o sistema informático, se pueden distinguir *3 fases* básicas:

- **Diseño:** en esta fase se elige el tipo de red neuronal a usar (la *arquitectura* o *topología*), el número de neuronas que la compondrán.
- **Entrenamiento:** en esta fase se le presentan a la red neuronal una serie de datos de entrada y datos de salida (resultados), para que a partir de ellos pueda *aprender*.
- **Uso:** se le suministran las entradas pertinentes a la red, y esta genera las salidas en función de lo que ha aprendido en la fase de entrenamiento.

➤ FUNCIONAMIENTO BASICO

Las *redes neuronales* están formadas por un conjunto de **neuronas artificiales** interconectadas.

Las neuronas de la red se encuentran distribuidas en diferentes **capas** de neuronas, de manera que las neuronas de una capa están *conectadas* con las neuronas de la capa siguiente, a las que puede enviar *información*.

La arquitectura más usada en la *actualidad* de una red neuronal consistiría en:

- Una primera *capa de entradas*, que recibe información del *exterior*.
- Una serie de *capas ocultas (intermedias)*, encargadas de realizar el trabajo de la red.
- Una *capa de salidas*, que proporciona el resultado del trabajo de la red al *exterior*.

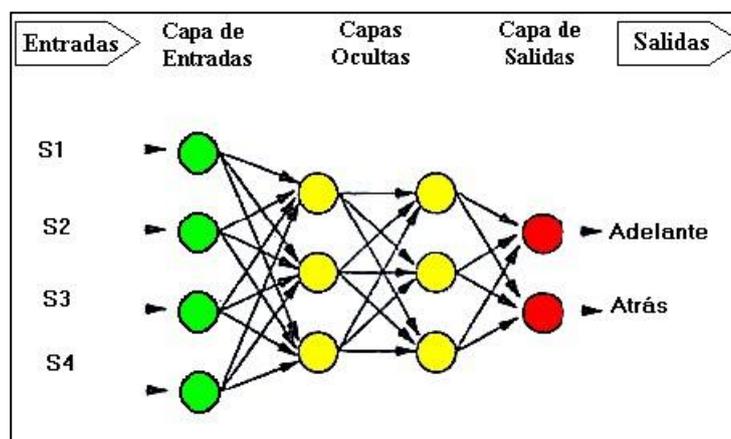


Figura 6.1 Esquema de una red neuronal antes del entrenamiento. Los círculos representan neuronas mientras las flechas representan conexiones entre las neuronas.

El número de capas intermedias y el número de neuronas de cada capa dependerá del tipo de aplicación al que se vaya a destinar la red neuronal.

➤ NEURONAS y CONEXIONES

Cada neurona de la red es una **unidad de procesamiento** de información; es decir, recibe información a través de las conexiones con las neuronas de la capa anterior, procesa la información, y emite el resultado a través de sus conexiones con las neuronas de la capa siguiente, siempre y cuando dicho resultado supere un valor "*umbral*".

En una red neuronal ya *entrenada*, las *conexiones* entre neuronas tienen un determinado peso ("*peso sináptico*").

Un *ejemplo* de una neurona sobre la que convergen conexiones de diferente peso sináptico (W_i) sería el de la **figura 6.2**.

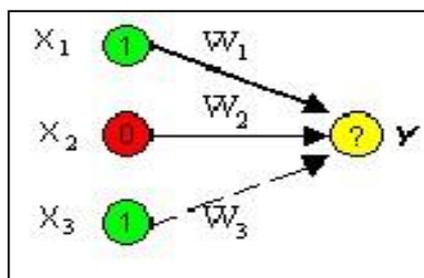


Figura 6.2 Conexiones de diferente peso sináptico ($W_1 > W_2 > W_3$) convergen sobre la misma neurona Y.

El *procesamiento de la información* llevado a cabo por cada neurona Y, consiste en una *función* (F) que opera con los *valores recibidos* desde las neuronas de la capa anterior (X_i , generalmente 0 o 1), y que tiene en cuenta el *peso sináptico* de la conexión por la que se recibieron dichos valores (W_i). Así, una neurona dará mas importancia a la información que le llegue por una conexión de peso mayor que no a aquella que le llegue por una conexión de menor peso sináptico.

Un modelo simple de la *función* F sería:

$$F = X_1W_1 + X_2W_2 + \dots + X_iW_i$$

Si el *resultado* de la función F es *mayor* que el valor *umbral* (U), la neurona se *activa* y emite una *señal* (1) hacia las neuronas de la capa siguiente. Pero, si por el contrario, el *resultado* es *menor* que el valor *umbral*, la neurona permanece *inactiva* (0) y no envía ninguna señal:

$$X_1W_1+X_2W_2+\dots+X_iW_i \leq U \text{ Inactivación } Y=0$$

$$X_1W_1+X_2W_2+\dots+X_iW_i > U \text{ Activación } Y=1$$

De esta forma, definido un conjunto inicial de pesos en las conexiones, al presentar un "estímulo" (conjunto de ceros y unos que representa un dato, perfil u objeto) a la *capa de entradas*, cada neurona en cada capa realiza la *operación* descrita anteriormente, activándose o no, de manera que al final del proceso las neuronas de la *capa de salidas* generan un resultado (otro conjunto de ceros y unos), que puede coincidir o no con el que se desea asociar el estímulo.

En el *entrenamiento* de una red neuronal tanto el *peso sináptico* de las conexiones como el *valor umbral* para cada neurona se modifican (según un algoritmo de aprendizaje), con el fin de que los resultados generados por la red *coincidan* con (o se aproximen a) los resultados esperados.

Y para simplificar el sistema de entrenamiento, el valor *umbral* (U) pasa a expresarse como un peso sináptico más ($-W_0$), pero asociado a una *neurona siempre activa* (X_0). Esta *neurona siempre activa*, se denomina "*bias*", y se sitúa en la capa anterior a la neurona Y .

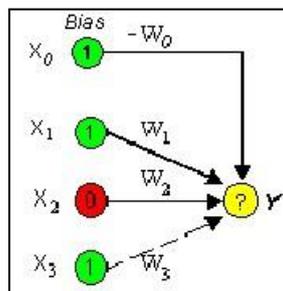


Figura 6.3 Neurona bias y su peso sináptico asociado ($-W_0$) en sustitución al valor umbral

Así, la condición de activación puede reescribirse como:

$$X_0 W_0 + X_1W_1+X_2W_2+\dots+X_iW_i > 0 \text{ Activación } Y=1$$

De esta manera el algoritmo de aprendizaje puede ajustar el *umbral* como si ajustara un *peso sináptico* más.

➤ TIPOS DE APRENDIZAJE BASICOS

Para poder aprender, las redes neuronales se sirven de un *algoritmo de aprendizaje*.

Estos algoritmos están formados por un *conjunto de reglas* que permiten a la red neuronal *aprender* (a partir de los *datos* que se le suministran), mediante la modificación de los *pesos sinápticos* de las conexiones entre las neuronas (recordar que el umbral de cada neurona se modificará como si fuera un peso sináptico más).

Generalmente los *datos* que se usan para entrenar la red se le suministran de manera aleatoria y secuencial.

Los *tipos de aprendizaje* pueden dividirse básicamente en tres, atendiendo a como esta *guiado* este aprendizaje:

- **Aprendizaje supervisado:** se introducen unos valores de entrada a la red, y los valores de salida generados por esta se *comparan* con los valores de salida correctos. Si hay *diferencias*, se *ajusta* la red en consecuencia.
- **Aprendizaje de refuerzo:** se introducen valores de entrada, y lo único que se le indica a la red si las salidas que ha generado son *correctas* o *incorrectas*.
- **Aprendizaje no supervisado:** *no* existe ningún tipo de *guía*. De esta manera lo único que puede hacer la red es reconocer *patrones* en los datos de entrada y crear *categorías* a partir de estos patrones. Así cuando se le entre algún dato, después del entrenamiento, la red será capaz de clasificarlo e indicará en que categoría lo ha clasificado.

➤ UN MODELO SIMPLE DE RED NEURONAL

Consideremos una red neuronal formada por *2 capas*:

- Una **capa de entradas** formada por 2 neuronas: **1** y **2**.
- Una **capa de salidas** formada por una sola neurona: **3**.

Las conexiones entre las dos neuronas de entrada y la neurona de salida presentan *pesos sinápticos ajustables* mediante el entrenamiento. Y a su vez, el valor *umbral* (U) de la neurona 3 puede ser ajustado como un peso sináptico más, al considerar $U = -W_0$ asociado a una *neurona bias* (siempre activa: $X_0 = 1$).

El *esquema* de este modelo sería la presente siguiente figura:

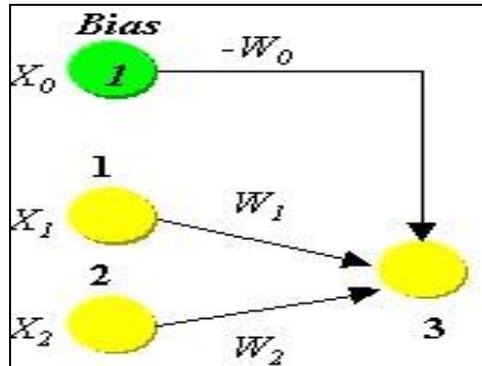


Figura 6.4 Modelo simple formado por tres neuronas(1,2 y 3), más una neurobias

Y la condición de activación de la neurona 3 sería:

$$X_0 W_0 + X_1 W_1 + X_2 W_2 > 0 \text{ Activación}$$

Además, en este modelo, cada neurona de una capa "transmite" su estado de activación (0 o 1) a la siguiente capa de neuronas, y después deja de estar activa.

Ahora, entrenaremos la red de manera que la neurona 3 se *active* sólo si las neuronas 1 y 2 están *activas*, y permanezca *inactiva* en caso *contrario*. Hay así cuatro posibles casos:

	X ₁	X ₂	CAPA DE SALIDA (X ₃)
Caso 1	0	0	0
Caso 2	0	1	0
Caso 3	1	0	0
Caso 4	1	1	1

Tabla 6.6 Valores entrada y salida

Las *conexiones* entre las neuronas de entrada *activas* y las neuronas de salida *activas* se *refuerzan* durante el entrenamiento: *coincidencias* entre actividad de entrada y actividad de salida se intensifican. Mientras que las conexiones entre neuronas de entrada inactivas y neuronas de salida (activas o inactivas) no se refuerzan.

➤ ENTRENAMIENTO DE LA RED

Debido al problema de la regla de aprendizaje de Hebb, usaremos una *regla de aprendizaje* derivada, y que si tenga en cuenta la *eficacia* de la red en cada momento:

1. Si la salida generada por la neurona 3 (para los valores de las otras dos neuronas) es la *correcta*, **no** se realizan ajustes de los pesos sinápticos.
2. Si la salida es **1** pero debería ser **0**, se **reducen** sólo los pesos de las conexiones activas según una constante **C**.
3. Si la salida es **0** pero debería ser **1**, entonces se **aumentan** sólo los pesos de las conexiones activas según la misma constante **C**.

Así, en cada ciclo de entrenamiento:

- Se *presenta* un dato de *entrada* (formado por los valores de las neuronas 1 y 2) del conjunto de datos de entrenamiento.
- La red, a partir del dato de entrada generará un dato de *salida*.
- Se aplica la *regla* anterior, que mide la eficiencia de la red y actúa en consecuencia.

Y se realizan diferentes ciclos de entrenamiento con los valores de entrenamiento hasta que la red responda correctamente a todos los datos de entrada (en todos los casos de entrenamiento) o hasta que la red ya no pueda aprender más.

De esta forma, si partimos de unos pesos sinápticos iguales a 0 ($W_1 = W_2 = W_0 = 0$) y de una constante de refuerzo igual a 1 ($C = 1$):

- Introducimos el dato de *entrada* del **caso 1**: $(X_1, X_2) = (0, 0)$.
- Puesto que todos los pesos valen 0:

$$X_0 W_0 + X_1 W_1 + X_2 W_2 = 0 \text{ Inactivación}$$

- Así el valor de *salida* de la neurona 3 es 0, y como el valor de la neurona 3 tenía que ser 0 (según el caso 1 de entrenamiento), *no* se realizan *ajustes* de los pesos.
- Introducimos el dato de *entrada* del **caso 4**: $(X_1, X_2) = (1, 1)$.
- Puesto que todos los pesos valen 0:

$$X_0 W_0 + X_1 W_1 + X_2 W_2 = 0 \text{ Inactivación}$$

- Pero, como el valor de *salida* de la neurona 3 es 0, y tenía que ser 1 (según el *caso 4* de entrenamiento), los pesos de las conexiones activas se *aumentan* en un valor de C (tal como indica el apartado 3 de la regla de aprendizaje):

$$W_1 = W_1 + C = 0 + 1 = 1$$

$$W_2 = W_2 + C = 0 + 1 = 1$$

$$W_0 = W_0 + C = 0 + 1 = 1$$

Y se continuaría de esta manera hasta terminar el entrenamiento, en cuyo caso obtendríamos los siguientes *pesos sinápticos ajustados*:

PESO SINAPTICO	VALOR
W_1	1
W_2	2
W_0	-2

Tabla 6.7 Pesos sinápticos

➤ MODELO DEL PERCEPTRON

Este modelo de red simple, sería un modelo *reducido* del modelo del Perceptrón, desarrollado por *Rosenblatt* entre 1958 y 1962, a partir de los modelos de red de *McCullough-Pitts*.

En el *ojo*, la imagen proyectada sobre la retina es convertida en señales eléctricas gracias a las células sensitivas a la luz (conos y bastones). Estas señales, a través de los axones de las células ganglionares, son transmitidas al cerebro, que interpreta la imagen.

El *modelo* del Perceptrón está basado en esta idea del *ojo*: las células sensibles de la retina serían *detectores* conectados con neuronas de la *capa de entrada*; estas neuronas se activan (1) cuando se activa el detector correspondiente. Cada neurona de la capa de entrada está conectada con diferentes neuronas de la *capa de detección de características*. Y cada neurona de esta capa estaría conectada a su vez con diferentes neuronas de la *capa de salidas* del Perceptrón.

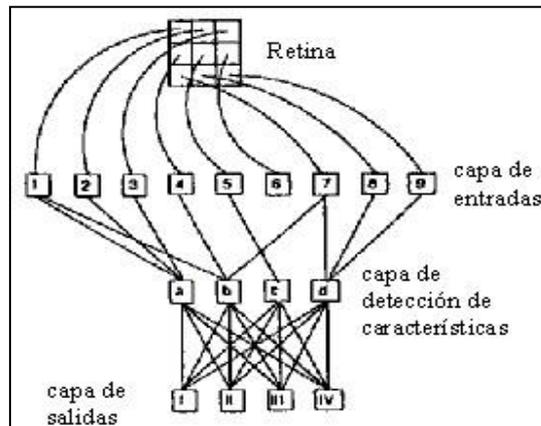


Figura 6.5 Modelo del Perceptrón

Y la regla de aprendizaje usada para entrenar la red es una versión simplificada de la regla de aprendizaje del Perceptrón.

➤ LIMITACIONES DEL MODELO

El modelo simple de red neuronal (al igual que el modelo del Perceptrón en el que se basa), presenta bastantes limitaciones en las tareas que puede llegar a aprender.

Así, por ejemplo, el modelo sería incapaz de aprender a realizar la operación *XOR* (*O* exclusivo): no puede ser entrenado con éxito para que la neurona 3 se active si las neuronas 1 o 2 están activas, pero *no* las dos a la vez.

Esto es debido a que el método de entrenamiento lo que hace es mover la siguiente recta sobre un plano:

$$X_1W_1 + X_2W_2 - U = 0$$

De manera que los puntos (X_1, X_2) del plano por encima de la recta hacen que la neurona 3 se active, y los que quedan por debajo de la recta hacen que la neurona 3 permanezca inactiva:

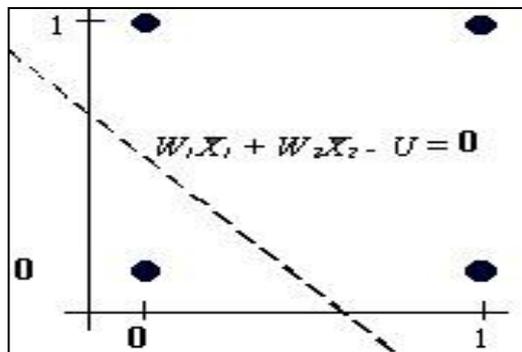


Figura 6.6 Plano de Activación dividido por la recta marca la condición de activación

Y no se puede dividir el plano en 2 partes de manera que solo los puntos (0,1) y (1,0) queden en la zona de activación de la neurona 3.

Para realizar este tipo de tareas "complejas" es preciso una red neuronal que contenga más de una capa de pesos sinápticos ajustables. Para ello, se añaden *capas ocultas* de neuronas entre la capa de entradas y la capa de salidas.

6.6.3. TECNICAS DE SOLUCION DE PROBLEMAS EN IA

Las técnicas de solución de problemas en IA, en general, incorporan un proceso de búsqueda. Todo proceso de búsqueda puede ser visualizado como el recorrido por un árbol en el que cada nodo representa un estado y cada rama representa las relaciones entre los estados cuyos nodos conecta.

En general, las reglas contienen en forma implícita el árbol, y se genera en forma explícita sólo aquellas partes que se decide explorar. Las principales diferencias que pueden aparecer en las diferentes técnicas de búsqueda, son:

- La dirección en la cual se conduce la búsqueda (hacia adelante o hacia atrás).
- La estrategia de control, o forma de seleccionar las reglas que pueden ser aplicables. Los principales requerimientos de una buena estrategia de control son: que cause desplazamiento en el espacio de estado; y, que sea sistemático.
- La forma de representar cada nodo del proceso de búsqueda (representación del conocimiento).

Muchas veces, tratar el proceso como búsqueda en un grafo en lugar de una búsqueda en un árbol, puede reducir el esfuerzo que se gasta en explorar senderos, esencialmente iguales, varias veces. Sin embargo, los requisitos asociados, son:

- Cada vez que se genere un nodo se debe chequear para ver si ha sido generado antes.
- Se deben introducir procedimientos especiales para que la búsqueda no quede atrapada en algún lazo.

A continuación se describen los algoritmos de tres procesos básicos de búsqueda de soluciones en el espacio de estado.

Algoritmo Generación Y Prueba (GENERATE-AND-TEST)

1. Generar una posible solución. (estado o camino)
2. Comprobar para ver si es una solución, mediante comparación con los elementos del conjunto de objetivos aceptables.
3. Si la solución ha sido encontrada salir, de otra manera, retornar al paso 1.

Algoritmo Primero a lo Ancho (BREATH-FIRST)

1. Crear una variable NODE_LIST y ponerla al estado inicial.
2. Hasta que se encuentre el objetivo o hasta que NODE_LIST esté vacía haga lo siguiente:

Remover el primer elemento de NODE_LIST, y llamarlo E. Si NODE_LIST estuvo vacía, salir.

Para cada forma en que cada regla puede ajustarse al estado descrito en E, haga lo siguiente:

- i. Aplicar la regla para generar un nuevo estado.
- ii. Si el nuevo estado es un estado objetivo, salir y retornar este estado.
- iii. Sino, añada el nuevo estado al final de NODE_LIST.

Algoritmo Primero en Profundidad (DEPTH-FIRST)

1. Si el estado inicial es el objetivo, salir y retornar éxito.
2. Sino, haga lo siguiente hasta que se obtenga señal de éxito o fracaso:
 - a) Genere un sucesor E del estado inicial. Si no hay más sucesores, retorne con señal de fracaso.
 - b) Llame recursivamente al algoritmo, esta vez con E como el estado inicial.
 - c) Si la señal es éxito, retorne, de otra manera, continúe en este lazo.

6.7. MANEJO DE INCERTIDUMBRE

En situaciones reales, no siempre es posible contar con toda la información, inclusive la información disponible puede ser incorrecta, incompleta o cambiar muy rápidamente. Todo esto da lugar a diferentes formas de inconsistencia e incertidumbre.

Diversos métodos han sido desarrollados para evaluar los grados de certeza o de verdad de las conclusiones. Uno de los más generalizados consiste en asignar coeficientes (pesos) de certeza o de confianza a los hechos que intervienen en las condiciones y en la conclusión de una regla.

Los principales modelos desarrollados son:

- Modelo estadístico - probabilístico.
- Modelo aproximado.

6.7.1. RAZONAMIENTO ESTADISTICO PROBABILISTICO

La técnica más antigua y mejor definida para manejar la incertidumbre es la **Regla de Bayes**, la misma que está basada en la teoría clásica de la probabilidad. Las hipótesis son más o menos probables dependiendo de las posibilidades de los hechos o evidencias que las sostienen. Las probabilidades se calculan en base a la fórmula general de la probabilidad condicionada de *Bayes* o alguna transformación de la misma.

El procedimiento para el modelo probabilístico es el siguiente:

- El factor de un conjunto de condiciones unidas por el operador lógico Y (AND) es igual al producto de cada una de las evidencias que intervienen.

- El factor de un conjunto de condiciones unidas por el operador lógico O (OR) es igual al complementario del producto de los complementarios de cada una de las evidencias que intervienen.
- Para el cálculo del coeficiente de la regla se aplica la Regla de Bayes:

A pesar que el método de Bayes es mucho más desarrollado que otros métodos para manejar incertidumbre, no deja de tener ciertas dificultades prácticas:

- Requiere de una gran cantidad de datos probabilísticos para construir una base de conocimientos. Por ejemplo, si un sistema de diagnóstico posee p conclusiones detectables y q características observables relevantes, requiere un mínimo de $(p * q + p)$ valores probabilísticos, asumiendo que: todas las conclusiones son mutuamente excluyentes, las características son condicionalmente independientes para cada conclusión, y que todas las características son valores verdaderos. Caso contrario, se requeriría de un número significativamente mayor que el indicado.
- Los tamaños de la muestra para obtener las probabilidades condicionales deben ser lo suficientemente grandes, como para que las probabilidades obtenidas sean exactas y significativas.
- A menudo las relaciones entre la hipótesis y la evidencia son importantes para determinar la forma en que la incertidumbre será manejada. Al reducirse estas asociaciones a simples números, remueve información relevante que podría utilizarse para razonar con éxito acerca de las incertidumbres.
- La reducción de dichas asociaciones a números también elimina la posibilidad de utilizar este conocimiento en otras tareas.

6.7.2. INFERENCIA APROXIMADA

Debido a la limitación que se tiene en la mayoría de los casos prácticos de no disponer de una gran cantidad de datos históricos, obligó al desarrollo de un método de inferencia aproximado, que en general proporcione resultados exactos aún cuando estén basados en datos limitados. A este se lo denominó el formalismo del **factor de certeza**.

Los resultados son más o menos ciertos en función de la certeza o falsedad de los hechos y conocimientos utilizados. Los factores de certeza que van asociados a los predicados, por lo general pueden ir de 0 a 100. El procedimiento a seguirse en el modelo aproximado es el siguiente:

- El factor de un conjunto de condiciones unidas por el operador lógico Y (AND) es igual al mínimo de los factores que intervienen.
- El factor de un conjunto de condiciones unidas por el operador lógico O (OR) es igual al máximo de los factores que intervienen.
- El factor de certeza de una conclusión es igual al producto del factor de certeza de las condiciones por el de la regla.

Para el cálculo del coeficiente resultante de la totalidad de reglas que se han encadenado en la deducción de un hecho, se aplican la siguiente relación:

- El formalismo del factor de certeza es uno de los más populares en los sistemas basados en conocimientos, debido a los siguientes puntos:
 - Es un modelo computacional simple que permite a los expertos estimar la confianza en las conclusiones derivadas.
 - Permite expresar el grado de creencia en cada hipótesis, permitiendo la expresión del efecto de múltiples fuentes de evidencia.
 - Permite la captura de conocimiento en una regla, incluyendo la cuantificación de la incertidumbre.
 - La asignación de valores de certeza es simple aunque subjetiva. No se requiere base estadística, simplemente se solicita al experto su estimación.
- También han sido puntualizadas ciertas dificultades asociadas a los factores de certeza:
 - Los valores de certeza no pueden representar en forma eficiente y natural ciertas dependencias entre estimaciones de incertidumbre.
 - El valor del factor de certeza asociado a una regla es dependiente de la fuerza de la asociación entre las premisas y las conclusiones. A medida que mayor conocimiento sea descubierto y añadido o retirado de una base de conocimiento, obligará a que los valores de certeza del conocimiento existente tengan que variar.

6.8. NOTAS BIBLIOGRAFICAS

Inteligencia, es la aptitud de crear relaciones. Esta creación puede darse de manera puramente sensorial, como en la inteligencia animal; también puede darse de manera intelectual, como en el ser humano.

El conocimiento puede ser representado como imágenes mentales en nuestros pensamientos, como palabras habladas o escritas en algún lenguaje, en forma gráfica o en imágenes, como cadenas de caracteres o colecciones de señales eléctricas o magnéticas dentro de un computador.

Inferir es concluir o decidir a partir de algo conocido o asumido. Por otro lado, razonar es pensar coherente y lógicamente a partir de hechos conocidos o asumidos.

Existen varias técnicas heurísticas buenas, de propósito general, que son útiles para una diversidad de problemas. Adicionalmente, es posible construir heurísticos especiales que exploten conocimiento específico en cada dominio, para resolver problemas particulares.

📖 LIBROS

- **ROLSTON, DAVID** (1985), "Principios de Inteligencia Artificial y Sistemas Expertos"
- **WINSTON, PATRICK** (1994), "Inteligencia Artificial", Tercera edición, Wilinington, EE.UU.

🌐 INTERNET

- www.insumostecnologicos.com.mx
- www.thi.com.ar
- www.inferencia.com

CAPITULO VI	110
INTELIGENCIA DE LOS ROBOTS	110
6.1. FUNDAMENTOS DE LA INTELIGENCIA ARTIFICIAL	111
6.2. DEFINICIONES DE INTELIGENCIA ARTIFICIAL	112
6.3. IMPORTANCIA DE LA INTELIGENCIA ARTIFICIAL	113
6.4. PROBLEMAS Y TECNICAS EN INTELIGENCIA ARTIFICIAL	114
6.4.1. CONOCIMIENTO.....	114
6.5. REPRESENTACION DEL CONOCIMIENTO	115
6.5.1. LOGICA PROPOSICIONAL.....	116
6.5.2. LOGICA DE PREDICADOS.....	118
6.5.3. REGLAS DE PRODUCCION.....	121
6.5.4. REDES ASOCIATIVAS.....	125
6.5.5. MEDIANTE PLANTILLAS.....	125
6.5.6. MEDIANTE OBJETOS.....	126
6.6. MANIPULACION DEL CONOCIMIENTO	126
6.6.1. PROCESOS HEURISTICOS DE BUSQUEDA.....	128
6.6.2. REDES NEURONALES.....	130
6.6.3. TECNICAS DE SOLUCION DE PROBLEMAS EN IA.....	140
6.7. MANEJO DE INCERTIDUMBRE	142
6.7.1. RAZONAMIENTO ESTADISTICO PROBABILISTICO.....	142
6.7.2. INFERENCIA APROXIMADA.....	143
6.8. NOTAS BIBLIOGRAFICAS	145