

INTRODUCCION AL APRENDIZAJE DE LA MAQUINA: SISTEMAS DISCENTES

1. INTRODUCCION

El aprendizaje es un término que cobija muchas facetas, por ejemplo:

- Adquisición de conocimientos "declarativos"
- Desarrollo de capacidades motoras y cognoscitivas gracias a la instrucción y la práctica.
- Organización de conocimientos en representaciones generales y efectivas.
- Descubrimiento de nuevos hechos y teorías gracias a la observación y experimentación.

Por su importancia fundamental como parte de la inteligencia, el estudio del aprendizaje constituye uno de los temas claves de la investigación en I. A. Se pretende tanto comprender los procesos de aprendizaje como implantarlos en sistemas artificiales.

En la investigación de sistemas discentes se pueden distinguir 3 tendencias :

- 1- Estudios orientados a tareas específicas:
Sistemas que mejoren su desempeño para determinada tarea, como resultado de la experiencia.
- 2- Modelación de los procesos de aprendizaje humanos o biológicos en general.
- 3- Análisis teóricos generales:
Investigación de posibles métodos de aprendizaje independientemente del dominio de aplicación y de su similitud o diferencia respecto a procesos biológicos.

2. MOTIVACION CIENTIFICO-FILOSOFICA

La cuestión de cuáles capacidades del ser humano u otro sistema biológico son innatas o genéticas y cuáles son aprendidas ha fascinado a biólogos, sicólogos, filósofos e investigadores de I. A. Una candidata a "capacidad innata" humana es el mecanismo de aprendizaje, o sea la capacidad de adquirir hechos, habilidades y conceptos abstractos. Por esto, el entender el aprendizaje humano lo suficiente para reproducir aspectos de él en una máquina es en sí mismo un objetivo científico válido. Adicionalmente, los modelos computacionales ayudan a la psicología cognoscitiva para forzarla a concretar sus teorías evitando que carezcan realmente de sentido, o sean tautológicas o que no se puedan probar.

El estudio de los proceso de aprendizaje en general, y del aprendizaje humano en particular, puede conducir a mejores técnicas educacionales. Por esto es natural que la investigación sobre Instrucción Asistida por Computador y la investigación sobre Sistemas Discentes comparten objetivos. Un desarrollo interesante en este sentido son los sistemas tutoriales por computador

que incorporan modelos del estudiante a partir del comportamiento que la máquina observa de este, permitiendo una individualización óptima del proceso de enseñanza.

También, como no hay demostración ni razón para creer que los métodos de aprendizaje humanos sean los únicos medios posibles de adquirir conocimientos y habilidades; la investigación sobre Sistemas Discentes permite explorar libremente mecanismos de aprendizaje alternativos teniendo como criterio su generalidad, utilidad y desempeño, sin importar su fidelidad a modelos psicológicos o biológicos. Esta investigación incluye:

- Diferentes mecanismos de inducción.
- Alcance y limitaciones de los métodos.
- Información de que debe disponer el aprendiz.
- Maneras de vérselas con el ruido y los datos de entrenamiento imperfectos.
- Desarrollo de técnicas generales, aplicables a muchos dominios diversos.

3. MOTIVACION PRAGMATICO-TECNOLOGICA

La investigación sobre máquinas discentes responde a una necesidad práctica, puesto que la automatización de un proceso requiere demasiado trabajo en dos fases:

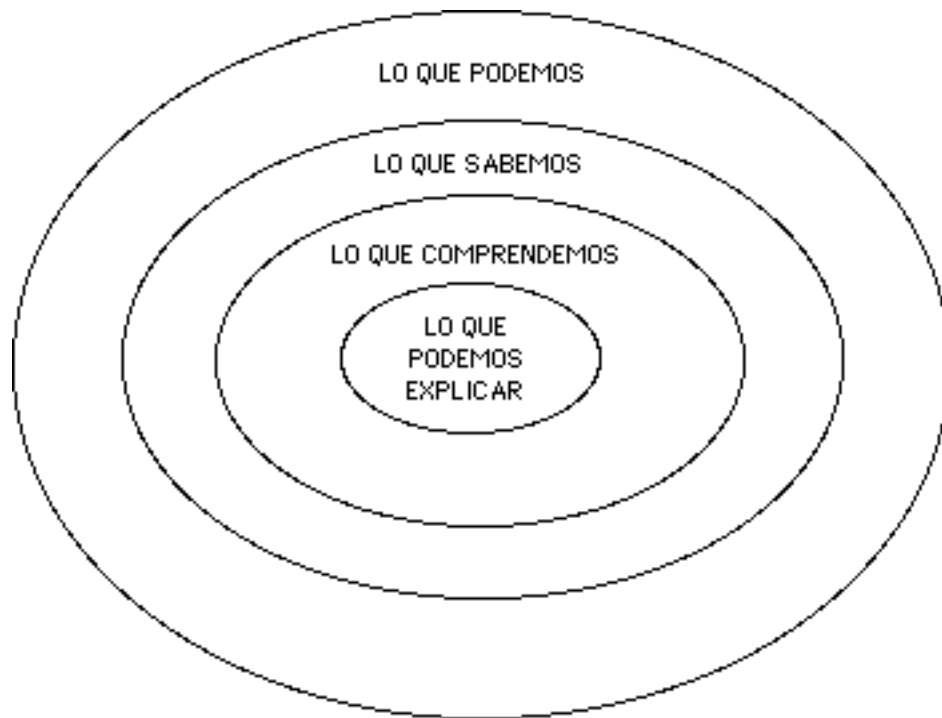
- 1- Para comprenderlo a tal punto de poder formular algoritmos para su solución.
- 2- Para programar explícitamente dichos algoritmos en la máquina.

Los computadores y robots actuales no pueden, o tienen muchas limitaciones en cuanto a:

- Aprender a hacer una tarea basándose en ejemplos o en la analogía con tareas similares previamente resueltas.
- Mejorar en su desempeño con base en errores anteriores.
- Adquirir nuevas habilidades observando e imitando a los expertos.

La investigación sobre el aprendizaje, dentro de la I. A. busca dotar a las máquinas de estas capacidades para facilitar la *programación* haciéndola automática. Esto es particularmente útil en la preparación de Sistemas Expertos, y en las tareas en que no conocemos *algoritmos* para programarlos directamente.

Especialmente, en el trabajo de desarrollo de Sistemas Expertos, se reconoce que la tarea de *Adquisición de Conocimientos* es la etapa más laboriosa, y para la cual los sistemas discentes pueden aportar un alivio al liberar al Ingeniero del Conocimiento de parte de esa engorrosa labor. Existe dificultad para expresar el conocimiento poseído, porque, como lo expresó el fisiólogo y filósofo del método científico Claude Bernard: " Podemos más de lo que sabemos , sabemos más de lo que comprendemos , y comprendemos más de lo que podemos explicar " .



Y precisamente, para codificar los conocimientos del experto necesitamos poder explicar sus razones.

Los expertos frecuentemente no siguen el razonamiento "lineal" que aparece en los textos. Ellos tienden a reconocer "patrones" o "configuraciones", lo que les posibilita utilizar de manera muy rápida conocimientos compilados a través de la experiencia, sin necesidad de reflexionar sobre los pasos involucrados

El conocimiento del experto se almacena "subconscientemente", y esto agrega poder al experto, pero al mismo tiempo dificulta esa transferencia de conocimientos requerida para implementar una B. C. Un experto puede tener más dificultad que un principiante para explicar sus acciones debido a que mucho de lo que hace ya tiene sus razones en el subconsciente.

Precisamente por esta dificultad para poder poner los conocimientos de forma explícita, la posibilidad de que la máquina pueda adquirir directamente conocimientos que están implícitos en la solución de casos anteriores, y no formulados explícitamente, es sumamente atractiva para el desarrollo de Sistemas Basados en Conocimientos.

4. DEFINICIONES DE "APRENDER"

- "Adquirir el conocimiento de alguna cosa" (Larouse, Salvat)
- "Modificar el comportamiento a causa de la adquisición de conocimiento"
- "Hacer cambios en el sistema para que mejore con el tiempo"

comentario: El vino mejora con el tiempo y esto no se considera aprendizaje.

- "Hacer cambios en el sistema, adaptativos en el sentido de que lo capacitan para hacer la misma tarea o tareas similares más efectivamente la próxima vez" (Herbert Simon)

comentario: Para aplicar este criterio habría que conocer los objetivos del sistema, lo cual no siempre es posible, alguien puede aprender a aparentar algo de tal manera que a un observador que no conoce el objetivo real, puede parecerle que ese alguien no ha aprendido.

- "Hacer cambios útiles en nuestra mente" (Marvin Minsky)

comentario: Demasiado general y por lo tanto casi inútil.

- "Construir o modificar representaciones de lo que se experimenta" (Ryszard S. Michalski).

comentario: Esta definición se centra en la representación de algún conocimiento, el aspecto de alguna mejora en el desempeño viene como consecuencia, ya que una definición que tome ese desempeño como criterio sería difícil de aplicar en casos como "aprender a apreciar la belleza" y además habría que conocer los objetivos del sistema para poder juzgar si mejoró el desempeño.

5. ADQUISICION DE CONOCIMIENTOS y REFINAMIENTO DE HABILIDADES

Se pueden distinguir dos formas básicas de aprendizaje:

- **Adquisición de Conocimientos, y**
- **Refinamiento de Habilidades.**

La Adquisición de Conocimientos involucra: Adquisición de conceptos, sus significados, sus relaciones entre sí y con el mundo externo y modelos de comportamientos. Puede darse en multitud de representaciones desde modelos intuitivos, imágenes, y ejemplos hasta ecuaciones matemáticas y leyes. En este sentido decimos que alguien ha aprendido cuando:

- Su conocimiento explica un conjunto más amplio de situaciones.
- Es más exacto al predecir el comportamiento de algún ente.

La segunda forma de aprendizaje consiste en la mejora gradual de habilidades motoras o cognitivas, tales como el aprender a montar en bicicleta o a tocar un instrumento musical. En estos casos, el adquirir conocimientos, por ejemplo de un libro, tan solo puede ser una fase inicial, dándose la mayor parte del aprendizaje durante la práctica repetida y la corrección de desviaciones respecto al comportamiento deseado.

La primera forma de aprendizaje se da a un nivel más conciente que la segunda forma, en la que el subconciente juega un papel primordial. La mayor parte del aprendizaje humano es una mezcla de esos dos tipos.

La investigación clásica de la I. A. ha estado más dirigida al proceso simbólico, y por esto puede parecer más afín con el primer tipo de aprendizaje, mientras que el segundo tipo de aprendizaje se acerca más a un proceso no simbólico tal como el estudiado en proceso de control adaptativo. Sin embargo, en principio ambos tipos se pueden capturar en modelos simbólicos dada la gran generalidad de estos, aunque por razones de eficiencia para el segundo tipo de aprendizaje pueden resultar más prácticos modelos no simbólicos como las redes neuronales.

6. UNA RESEÑA HISTORICA DEL ESTUDIO DE SISTEMAS DISCENTES

En el decenio de 1920 I. P. Pávlov realiza sus famosos experimentos sobre el reflejo condicionado.

En 1949 D. O. Hebb publica su teoría del aprendizaje neuronal.

En 1950 W. K. Estes propone una reoría estadística del aprendizaje, y en 1955 R. R. Bush y F. Mosteller continúan en con este tipo de enfoque.

En 1953 W. Grey Walter da a conocer su teoría sobre el cerebro y sus realizaciones físicas, denominadas *tortugas* por su aspecto.

En 1958 J. Brown publica un modelo de la memoria a corto plazo, que incluye el proceso del decaimiento de la misma; F. Rosenblatt publica su modelo *Perceptron*, apoyándose en la teoría de Hebb, y , por su parte A. J. Angyan publica un modelo de adaptación neuronal.

En 1959 Arthur Samuel construyó un programa que mejoraba su desempeño en el juego de damas a partir de la experiencia.

En el decenio de 1960 L. D. Harmon estudia ciertos modelos de neuronas artificiales, e igualmente por esta época se publican trabajos sobre el aprendizaje de redes por parte de M. L. Minsky, O. G. Selfridge, W. S. McCulloch y N. Rashevsky. También aparecen los trabajos sobre sistemas adaptativos de G. Pask.

En 1970 Patrik Henry Winston hizo un programa que adquiría conceptos en el dominio de las construcciones con bloques, a partir de instrucción.

Hacia 1975 John R. Anderson desarrolla el sistema ACT que mejora sus habilidades para la prueba de teoremas de geometría.

En 1977 Douglas Lenat realiza el sistema AM que descubre conceptos matemáticos a partir de unos principios básicos y unas heurísticas que lo guían. Por la misma época B. G. Buchanan y sus colaboradores desarrollan el Meta-Dendral que descubre reglas de rompimiento de enlaces químicos, para interpretar exámenes de un espectrógrafo de masas.

Pat Langley y su grupo desarrollan los programas BACON que descubren leyes empíricas en el área de fisicoquímica, a partir de datos numéricos de experimentos.

Por 1980 Tom M. Mitchell, Ranan Banerji y otros desarrollan el programa LEX que adquiere heurísticas para la solución de problemas de integración simbólica.

Los anteriores ejemplos ya van siendo "clásicos" en la literatura por haber sido algo notorios en su momento, pero en los últimos diez años se han multiplicado las investigaciones y diversificado los enfoques, los métodos y áreas de aplicación; también en la actualidad se realizan con regularidad varios congresos internacionales sobre el aprendizaje de la máquina.

7. TAXONOMIAS DE SISTEMAS DISCENTES

Se pueden clasificar los Sistemas Discentes, según varias dimensiones o criterios, por ejemplo: Según las Estrategias de Aprendizaje utilizadas, Según la Representación del Conocimiento, Según paradigmas: Simbólico o Conexionista

7.1 SEGUN ESTRATEGIA DE APRENDIZAJE :

Por la cantidad de inferencia que el "aprendiz" hace sobre la información disponible, consideremos dos extremos: Ninguna inferencia y mucha inferencia.

Si programamos directamente un computador, sus "conocimientos" aumentan, pero no hay ninguna inferencia por parte de la máquina, todo el esfuerzo cognoscitivo está de parte del programador. En el otro extremo, si un sistema independientemente descubre nuevas teorías o inventa nuevos conceptos, entonces hace una cantidad sustancial de inferencias: está derivando conocimientos organizados a partir de experimentación y observación. Entre esos dos extremos de esta dimensión, un punto intermedio sería un estudiante determinando cómo resolver un problema matemático por analogía con otros similares ya resueltos en un libro de texto: requiere inferencia, pero menos que descubrir sin ayuda del texto o del profesor.

Aprendizaje "de memoria" o implantación directa de conocimientos. Ejemplos: Aprender por "ser programado", y por memorización de hechos y datos a lo "base de datos" clásica).

Aprendizaje a partir de Instrucción : O sea a partir de una fuente organizada de información, lo cual requiere ciertas transformaciones de la información y algo de inferencia.

Aprendizaje por Analogía : Por ejemplo una persona que nunca ha manejado un camión pero que maneja automóvil, puede adaptar esa habilidad para manejar el camión, otro caso sería la adaptación de un programa de computador para una tarea análoga a aquella para lo que fue inicialmente hecho el programa.

Aprendizaje a partir de Ejemplos : (caso especial del "Aprendizaje Inductivo"). Es un tipo de aprendizaje muy estudiado en I. A.

Dado un conjunto de ejemplos y, a veces, antiejesmplos, de un concepto el aprendiz induce una descripción general del concepto que describa todos los ejemplos y ninguno de los antiejesmplos. Se puede subclasificar: Según el "instructor" sea otro sistema, o el medio ambiente, o el aprendiz mismo; y también según haya solo ejemplos positivos o se incluyan también antiejesmplos durante la instrucción.

Aprendizaje por Observación y Descubrimiento (o "Aprendizaje No Supervisado): Se puede subclasificar según el grado de interacción con el ambiente, siendo los extremos: Observación Pasiva y Experimentación Activa.

7.2 SEGUN REPRESENTACION DE CONOCIMIENTOS ADQUIRIDOS

La Representación del Conocimiento puede ser de diversas formas, y el aprendizaje entonces corresponde a modificaciones de diversa índole según la naturaleza de la Representación del Conocimiento, Ej:

Parámetros de ecuaciones (ajuste de coeficientes o umbrales de elementos lógicos).

Parámetros físicos (conectividad sináptica entre neuronas).

Arboles de Decisión y Taxonomías (para discriminar o clasificar)

Garamáticas Formales.

Reglas de Producción.

Formalismos Lógicos.

Grafos y Redes.

Marcos.

Programas y Procedimientos (generan programas automáticamente, o secuencias de instrucciones para manipuladores robóticos)

Representaciones Múltiples.

7.3 SEGUN PARADIGMAS: SIMBOLICO y CONEXIONISTA

Entendiendo por "Simbólico" el modelo que trata los fenómenos de la inteligencia y el aprendizaje como una manipulación de Símbolos en un sistema de la clase de los computadores

digitales y su software, y entendiendo por "Conexionistas" el modelo que trata la inteligencia y el aprendizaje como el resultado de un proceso de señales (Ej: Redes Neuronales).

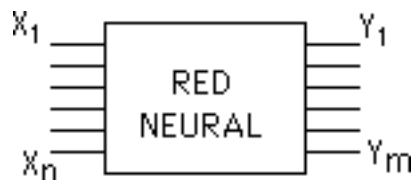
En el paradigma "Simbólico" el aprendizaje consiste en la creación y manipulación de símbolos, y en el "Conexionista" generalmente consiste en la modificación de parámetros como coeficientes de ponderación, niveles de umbral, o fuerza de la conexión entre elementos.

7.3.1 Algunas Técnicas de Aprendizaje Utilizadas en el Modelo Simbólico

- Modificación de descripciones de conceptos u objetos, estando estas descripciones representadas como redes de símbolos interrelacionados. En este caso el aprendizaje se hace modificando las conexiones o interrelaciones entre los símbolos.
- Modificación de "preferencias" de los diversos caminos de un árbol de decisión o búsqueda, según las experiencias que ha tenido el sistema.
- Modificaciones en las Representaciones Basadas en Reglas :
 - Adición de nuevas Reglas deducida a partir de casos resueltos.
 - Modificación de Reglas existentes (Generalización, Especialización, combinación de Reglas).
 - Modificación de las preferencias para el uso de las reglas, o de las Metareglas (Reglas sobre como usar las Reglas del dominio)

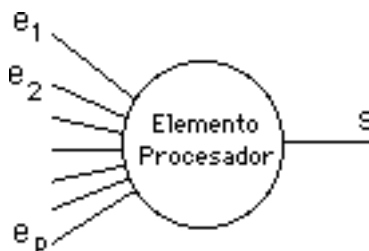
7.3.2 Redes Neuronales

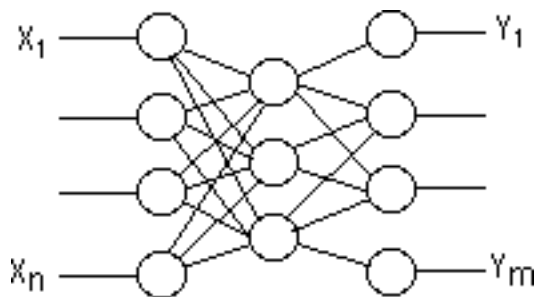
Paralelamente con la investigación típica de la I. A., que se centra en el desarrollo de modelos conceptuales y simbólicos, se han desarrollado sistemas análogos a las neuronas y sus redes, que resultan particularmente aptos para procesos de aprendizaje del tipo "mejora de habilidades" o "reconocimiento de patrones".



Una Red Neuronal realiza un "mapeado" continuo desde un espacio n-dimensional a uno m-dimensional, basándose en ejemplos de la acción del mapeado.

La Red Neuronal está compuesta internamente por muchos Elementos Procesadores, que, a la manera de las neuronas, están muy interconectadas, teniendo cada interconexión determinada fuerza.





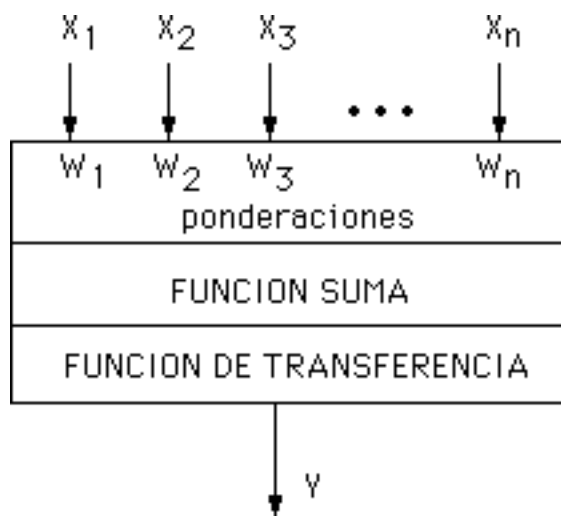
Durante el aprendizaje en este tipo de sistemas se ajustan los parámetros que definen las relaciones entre las entradas y salidas de cada elemento procesador.

Las Redes Neuronales se usan típicamente para desarrollar una relación entre varias entradas y varias salidas. Al someter la red a un conjunto de entradas y su correspondiente conjunto de salida, la red ajusta sus conexiones según cierta "regla de aprendizaje"

7.3.2.1. Especificación de una Red Neuronal Artificial

La especificación de un R. N. A. consta de la definición de tres componentes:

- 1 - ELEMENTO PROCESADOR ("neurona artificial", o "neurodo") : Especificación de la estructura del elemento procesador que es un nodo de la red. Asumimos por el momento que son iguales, se trata de una red con nodos homogéneos.



En esta especificación se establecen las funciones que relacionan las entradas a la neurona con las salidas, por ejemplo: Se puede componer de una primera función de SUMA o agregación:

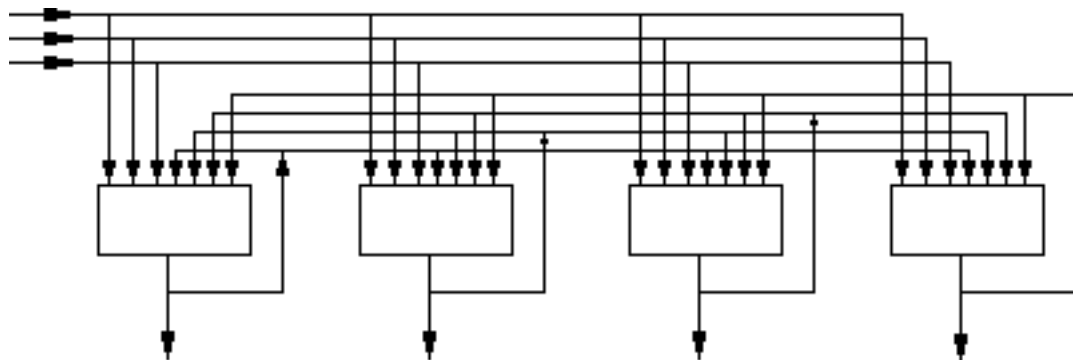
$S = W_0 + W_1X_1 + W_2X_2 + \dots + W_nX_n$ (siendo W_i = ponderaciones, X_i = entradas)

y una segunda función, de TRANSFERENCIA, por ejemplo :

$Y = (1 + e^{-s})^{-1}$ (siendo Y = salida del elemento)

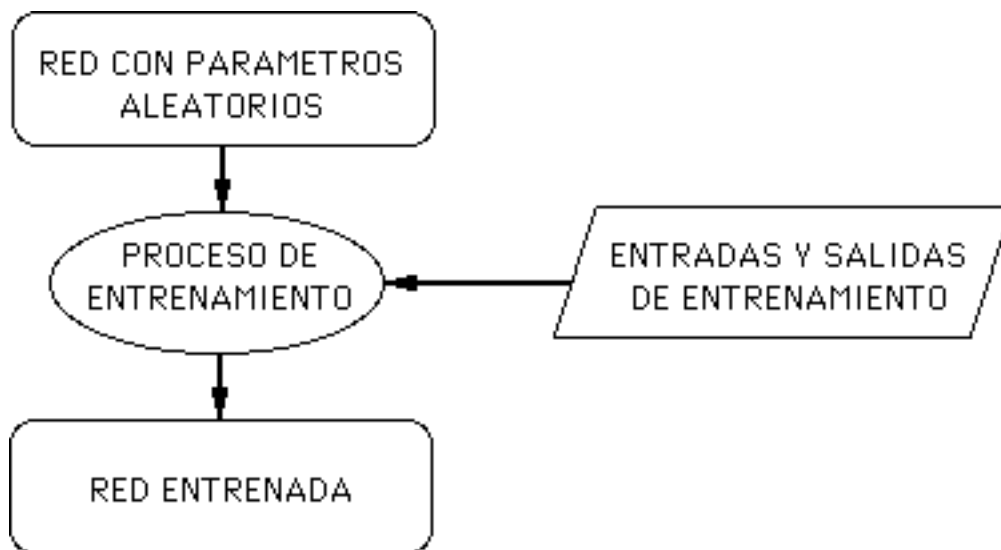
2- NEURODINAMICA : Especificación de las reglas de aprendizaje o adaptación, según las cuales la red ajusta parámetros internos W_i como resultado de la experiencia.

3 - TOPOLOGIA O ARQUITECTURA: Especificación de las interconexiones entre los elementos procesadores.

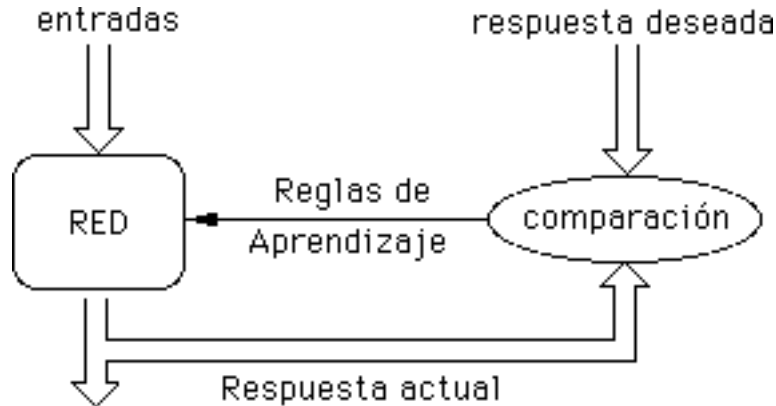


7.3.2.2. Entrenamiento de la Red

Inicialmente podemos tener la red configurada con unos parámetros de valores aleatorios, la cual sometemos a un período de aprendizaje, para obtener la red entrenada :



Durante el entrenamiento o aprendizaje la red modifica parámetros internos W_i basándose en la regla de aprendizaje, la cual tiene en cuenta la diferencia entre la respuesta que actualmente dió la red y la respuesta deseada.



7.3.2.3. La Red Neuronal como Sistema Experto

Realmente, las Redes Neuronales Artificiales (R.N.A) . no fueron inventadas para hacer sistemas expertos, aunque en cierto sentido se pueden comportar como estos. Las R.N.A. ofrecen pues una alternativa para implementar la función de un sistema experto.

Los principales pasos de implementación son los siguientes :

- 1 - Identificar todos los atributos A_i necesarios para resolver el problema, (ej: síntomas en un sistema de diagnóstico) .
- 2 - Identificar los valores V_{ij} asociados con cada atributo, o posibles características de los casos.
- 3 - Identificar las posibles conclusiones o selecciones .
- 4 - Escoger la red :
 - Tipo de elemento procesador.
 - Distribución de los elementos en capas de entrada, ocultas o intermedias, y de salida (inglés: "input layer", "hidden layer(s)", "output layer") .
 - Manera de codificar las características de los casos como valores de las entradas de los elementos de la capa de entrada .
 - Manera de codificar las selecciones de los casos como valores de las salidas de los elementos de la capa de salida .
 - Regla de aprendizaje, o sea el algoritmo según el cual se modificarán los pesos de interconexión entre los elementos.
- 5 - Utilizar un conjunto de casos ejemplo para entrenar la red . Los atributos y valores definen la capa de entrada de la red , mientras los resultados forman la capa de salida

Cuando se usan R.N.A. , el dominio y la tarea no requieren ser tan bien comprendidos como para usar los "shells" tradicionales. No se requiere que el experto detalle cómo llega a una solución, ya que el sistema se adapta con base en las relaciones estímulo/respuesta, reconfigurándose para resolver el problema . La ventaja principal de usar este tipo de implementación es que no requiere algoritmos específicos del dominio ; otra ventaja consiste en que se pueden degradar más "gracilmente" que los sistemas expertos tradicionales si falla alguna parte (por ejemplo una conexión) . La principal desventaja es que no tenemos manera

de examinar los "conocimientos" del sistema, y en este sentido es una especie de "caja negra", en el cual podemos confiar sólo por su comportamiento entrada/salida .

8. APRENDIZAJE DE CONCEPTOS A PARTIR DE EJEMPLOS

Un tipo de aprendizaje muy investigado en I.A. es el aprendizaje de conceptos a partir de ejemplos. Generalmente los conceptos se representan como un grafo que describe el concepto a ser aprendido, y a partir de ejemplos y antiejesmplos el sistema va modificando dicho grafo para que describa más precisamente todos los ejemplos del concepto, y en cambio no corresponda a ningún antiejesmplo.

En los sistemas que aprenden a partir de ejemplo se investiga actualmente :

- Cómo ampliar los tipos de representación, operadores y variables utilizados.
- Cómo lograr que los conceptos y generalizaciones a que llege el sistema sean muy comprensibles para los humanos, o sea que tengan en cuenta las limitaciones humanas para manejar, por ejemplo, una serie de preposiciones conectadas por "O" e "Y".
- Cómo guiar el mecanismo de inducción constructiva através del espacio, potencialmente inmenso, de posibles descriptores derivados.
- Buscando eficiencia, cómo incorporar, a un método general de inducción, conocimientos específicos del dominio de aplicación.
- La interfaz con el usuario : A medida que los programas discentes sean más perfectos, su funcionamiento tenderá a ser más opaco para el usuario. Es necesario que el sistema tenga dispositivos de explicación para justificar al usuario sus generalizaciones.
- Cómo funcionar bien en presencia de ruido : información incierta, incompleta o errónea.

9. INDUCCION AUTOMATICA PARA SISTEMAS EXPERTOS

9.1 INTRODUCCION

Entendiendo por "Simbólico" el modelo que trata los fenomenos de la inteligencia y el aprendizaje Para la tarea de **deducción** , conocemos reglas generales que podemos aplicar a casos particulares, por ejemplo, si tenemos como cierto que: *Todos los niños comienzan sus estudios primarios luego de cumplir siete años de edad* , y luego sabemos que: *María está en primaria* , podemos deducir que: *María tiene al menos siete años de edad* ; y similarmente, si se nos dice que *Juan tiene cinco años*, podemos deducir que: *Juan no está en primaria*.

La **inducción** trabaja en sentido contrario, desde los casos particulares hacia la regla general. Por ejemplo, a partir de la muestra de casos:

Nombre	Edad	En Primaria ?	
Mario	8	si	
Pedro	10	si	
David	5	no	
Santiago	9		si
Juliana	2		no
Alejandra	3		no

podemos inducir la regla: *Si el niño tiene al menos ocho años entonces está en primaria* . Realmente la edad podría ser seis, siete u ocho años, y la incertidumbre de la regla inducida se debe a los límites de la muestra de que disponemos.

La tarea de inducción consiste, en pocas palabras en hallar una *Regla de Clasificación o decisión*, a partir de los siguientes elementos:

- Un conjunto de *casos de ejemplo* de entrenamiento. Cada caso de ejemplo posee un conjunto de *valores* característicos de ciertos *atributos* (en el ejemplo anterior hay un solo atributo que es la edad).
- Un conjunto de *Clases*, que representan las categorías en que un "experto" clasifica cada caso ejemplo. (en el ejemplo anterior hay dos clases: En Primaria y No en Primaria).

La tarea de inducción se realiza ejecutando un *algoritmo inductivo*.

La calidad de lo inducido depende tanto del algoritmo inductivo como de la riqueza y representatividad del conjunto de casos ejemplo y sus atributos.

Los algoritmos inductivos se diferencian entre sí por la manera en la cual exploran en el conjunto de casos ejemplo, la manera como generalizan, y la manera como enfrentan los errores y el ruido en los casos entrados. Algunos algoritmos se diseñan pensando en un dominio específico, y otros pretenden ser de aplicación general. Por lo general, los de aplicación general pueden resultar más ineficientes y producir resultados menos espectaculares.

Ha habido una serie de resultados alentadores en cuanto a la inducción automática, por ejemplo, los investigadores Michalski y Chilausky estudiaron las enfermedades de la soya en una hacienda de Illinois, y produjeron dos conjuntos de reglas para el diagnóstico, el primero obtenido de un experto, y el otro inducido automáticamente a partir de ejemplos. Los ejemplos de entrenamiento consistían de descripciones físicas y ambientales de cada planta, junto con el diagnóstico de cada caso. En esta experiencia se comprobó que las reglas inducidas resultaron mejores que el experto para el diagnóstico de nuevos casos. El algoritmo utilizado en ese caso fue el conocido como AQ11. Algunas investigaciones sobre aprendizaje de máquina señalan que los algoritmos inducen un conjunto de reglas más simple que el que tienden a expresar los expertos.

9.2 ALGORITMO ID3

El investigador J. Ross Quinlan desarrolló el algoritmo conocido como ID3 (Iterative Dicotomiser 3), que es uno de los más utilizados y se encuentra implementado en paquetes de software comercial.

Aunque el algoritmo se desarrolló inicialmente para clasificar movimientos en ajedrez, con atributos categóricos (que podían tomar sólo valores discretos de un conjunto finito) y dos clases, la mayoría de las implementaciones actuales permiten atributos con valores enteros o reales, y clasificación en más de dos clases.

El ingeniero del conocimiento somete, como entrada al algoritmo, un conjunto de casos ejemplo en el cual debe haber representantes de todas las clases manejadas por el experto. El conjunto es analizado por el programa, el cual entrega las reglas inducidas.

Como se insinuó antes, la representatividad del conjunto de entrenamiento afecta la calidad de las reglas inducidas: el algoritmo es incapaz de descubrir algo que no esté allí. Por esto una buena selección de atributos y ejemplos es fundamental. La selección de atributos debe basarse en el conocimiento acumulado por la experiencia.

Los atributos son los factores que influyen en la clasificación o decisión. En este algoritmo cada atributo forma un nodo intermedio en un árbol cuyas hojas o nodos terminales son las clases o decisiones. Dado el conjunto de ejemplos, el ID3 selecciona el atributo que subdivide los ejemplos *de la mejor manera*. Un atributo categórico con **N** posibles valores subdividiría los ejemplos en **N** subconjuntos, y será *el mejor atributo* para la próxima regla si cada uno de los **N** subconjuntos contiene elementos de sólo una clase; sería un mal atributo a seleccionar si los subconjuntos poseen elementos de muchas clases. En el caso de atributos con valores numéricos se requiere escoger valores de corte (umbrales) de manera óptima para la subdivisión, por ejemplo "altura mayor de 51.2" divide los ejemplos en 2 subconjuntos. Como criterio para medir *el grado de mezcla de clases* en los subconjuntos, el ID3 utiliza una *medida de información*, basándose en la famosa teoría estadística de la información de Claude Shannon: selecciona el atributo que produce la mayor disminución de la *información* (en el sentido de Shannon) en los subconjuntos.

Si los atributos se terminan antes de completar el árbol, entonces hay ejemplos contradictorios en el conjunto de entrenamiento, o sea ejemplos con iguales valores de atributos, pero de clases diferentes. El algoritmo básico original no estaba diseñado para esto, y es necesario introducir un atributo extra para distinguir entre ejemplos contradictorios. En otras situaciones el proceso puede revelar carencias en el conjunto de entrenamiento, al identificar un nodo terminal en el árbol inducido, para el cual no hay un ejemplar, o sea que la clase para esa combinación de valores de atributos es desconocida.

La estructura del ID3 es iterativa. Se elige un subconjunto de los casos de entrenamiento, el cual se denomina *ventana*, y se forma un árbol de decisión a partir de ese subconjunto. Este árbol clasifica correctamente los casos de la *ventana*. Luego se intenta clasificar con ese árbol los restantes casos del conjunto de entrenamiento. Si el árbol da una correcta clasificación para todos los casos, entonces termina el proceso. Si no, entonces se seleccionan unos de los objetos incorrectamente clasificados, se agregan a la ventana y se repite el proceso, de esta manera, es usual hallar árboles de decisión correctos luego de unas pocas iteraciones para conjuntos de entrenamiento de hasta 30000 objetos, descritos por medio de hasta 50 atributos.

El árbol de decisión se recorre desde la raíz, y tanto en ella como en cada uno de los demás nodos se decide cuál rama tomar basándose en el valor de algún atributo del ejemplar que se esté clasificando, hasta llegar a un nodo terminal (hoja), que, como se mencionó antes, corresponde a la clase en que queda clasificado el ejemplar.

Suponiendo que el conjunto de casos de entrenamiento posee p ejemplares de clase P y n ejemplares de clase N, para el algoritmo ID3 asumimos lo siguiente:

- 1- Un árbol de decisión correcto clasificará los ejemplares, en cada clase, en las mismas proporciones en que están representados en el conjunto de entrenamiento. O sea que tomando un ejemplar al azar, este se clasificará como P con probabilidad $p/(p + n)$, y como clase N con probabilidad $n/(p + n)$.
- 2- El árbol de decisión se puede considerar como fuente de un mensaje: " P " o " N ", y la información esperada para generar este mensaje será entonces (con logaritmos base 2):

$$I(p, n) = -(p/(p + n)) * \text{LOG}(p/(p + n)) - (n/(p + n)) * \text{LOG}(n/(p + n))$$

Si usamos un atributo A, que tenga v valores posibles, como raíz del árbol, las v ramas salientes de la raíz conducen a su vez a v sub-árboles. Tomando un la i -ésima rama, que tiene p_i objetos clase P y n_i objetos clase N, la información requerida para ese i -ésimo sub-árbol será $I(p_i, n_i)$, y entonces la información esperada requerida para el árbol con A como raíz se obtiene como el promedio ponderado:

$$E(A) = ((p_1+n_1) * I(p_1, n_1) + (p_2+n_2) * I(p_2, n_2) + \dots + (p_v+n_v) * I(p_v, n_v)) / (p+n)$$

donde la ponderación para la i -ésima rama es la proporción de los objetos que pertenecen a esa rama, por lo tanto, la información ganada al ramificarse gracias al atributo A es:

$$\text{Ganancia}(A) = I(p, n) - E(A)$$

un buen criterio parece ser escoger el atributo que *gana* la mayor información. ID3 examina todos los atributos y elige el de máxima ganancia, forma la ramificación y usa el mismo proceso recursivamente para formar sub-árboles a partir de los v nodos generados

Para darnos una idea de la aplicación del algoritmo ID3, veamos un ejemplo. Sea el conjunto de entrenamiento el siguiente, que clasifica el estado del tiempo en dos clases, que denominaremos P y N, según 4 atributos:

Caso #	General	Temper.	Humedad	Viento	Clase
1	asoleado	caliente	alta	no	N
2	asoleado	caliente	alta	si	N
3	nublado	caliente	alta	no	P
4	lluvioso	templada	alta	no	P
5	lluvioso	fría	normal	no	P
6	lluvioso	fría	normal	si	N
7	nublado	fría	normal	si	P
8	asoleado	templada	alta	no	N
9	asoleado	fría	normal	no	P
10	lluvioso	templada	normal	no	P
11	asoleado	templada	normal	si	P
12	nublado	templada	alta	si	P
13	nublado	caliente	normal	no	P
14	lluvioso	templada	alta	si	N

Nueve objetos son clase P y cinco son clase N, entonces la información requerida para la clasificación es:

$$I(p, n) = - (9/14) * \text{LOG}(9/14) - (5/14) * \text{LOG}(5/14) = 0.940 \text{ bits}$$

Considerando el atributo `General`, con sus tres valores ($v=3$):

Para el primer valor, hay 5 objetos que lo tienen, 2 clase P y 3 clase N, entonces:

$$p_1 = 2, \quad n_1 = 3, \quad I(p_1, n_1) = 0.971$$

Análogamente, para el segundo valor posible de A :

$$p_2 = 4, \quad n_2 = 0, \quad I(p_2, n_2) = 0$$

Y para el tercer valor de A :

$$p_3 = 3, \quad n_3 = 2, \quad I(p_3, n_3) = 0.971$$

Por lo tanto el requisito de información esperada, después de chequear este atributo es:

$$E(\text{General}) = (5 \cdot I(p_1, n_1) + 4 \cdot I(p_2, n_2) + 5 \cdot I(p_3, n_3)) / 14$$

$$E(\text{General}) = 0.694$$

Y la Ganancia de este atributo es:

$$\text{Ganancia}(\text{General}) = 0.940 - E(\text{General}) = 0.246$$

Y el mismo procedimiento aplicado a los otros tres atributos da:

$$\text{Ganancia}(\text{Temperatura}) = 0.029$$

$$\text{Ganancia}(\text{Humedad}) = 0.151$$

$$\text{Ganancia}(\text{Viento}) = 0.048$$

Entonces ID3 escoge *General* como atributo para la raíz del árbol, y se repite el procedimiento para cada uno de los tres sub-árboles. Resulta así que en la rama de "asoleado" se pregunta luego por el atributo "Humedad", en la rama de "nublado" ya se llega a una clasificación como "P" sin necesidad de más chequeos, y en la rama de "lluvioso" se chequea el atributo "viento".

Aquí sólo se ha podido mostrar el fundamento del algoritmo ID3, pero este ha sido objeto de posteriores refinamientos, y además de él existen numerosos otros en continuo desarrollo.

Hemos visto pues como un algoritmo puede inducir un procedimiento que sirve para clasificar nuevos casos, a partir de ejemplos de clasificación (solución) de casos anteriores. El problema de la clasificación es del mismo tipo formal que muchos otros problemas de utilidad práctica, por ejemplo: problemas de diagnóstico, donde los valores de los atributos corresponden a "síntomas", y las "clases" corresponden a la falla diagnosticadas; problemas de decisión para asignar o no un crédito o expedir o no una tarjeta de crédito según los valores de "atributos económicos" del cliente, etc.

10. ALGORITMOS GENETICOS

Ciertos algoritmos que han sido inspirados en la evolución de las especies, tienen la propiedad de "adaptarse" encontrando soluciones a ciertos problemas que son óptimas según algún criterio preestablecido. Por su carácter adaptativo, merecen ser expuestas aquí, así sea brevemente.

Aunque la codificación de las características de una especie biológica en los cromosomas no se conozca en detalle, si hay una serie de hechos generalmente aceptados acerca de como funciona la evolución adaptativa de las especies, y que sirven para una aproximación a ese proceso que es susceptible de ser formalizada en algoritmos :

- La evolución es un proceso que opera sobre los cromosomas que codifican individuos, y no sobre los individuos.
- La Selección Natural es el enlace entre los cromosomas y el *desempeño* (performance) de sus correspondientes estructuras decodificadas (individuos). El proceso de selección natural hace que los cromosomas que codifican estructuras exitosas se reproduzcan más que los cromosomas de las que no lo son.

- La reproducción es el punto en el cual tiene lugar la evolución. La recombinación de la información genética puede crear cromosomas para los hijos muy diferentes de aquellos de los padres.
- Pueden ocurrir mutaciones que hagan que los cromosomas de los hijos difieran de los de los padres.
- La evolución no tiene memoria: lo que ella sabe acerca de producir individuos que funcionen bien en el ambiente está contenido en el conjunto de genes portados por los cromosomas de la población actual, y en la estructura de los decodificadores de cromosomas.

Estas características intrigaron a John Holland por el año 1970, y lo llevaron a plantear algoritmos que aplicaban esos principios, iniciando lo que luego se bautizó como **Algoritmos Genéticos**, y que luego contribuyó al campo denominado actualmente "Artificial Life" (más propiamente *Vida Formal*).

Una descripción de alto nivel de un algoritmo genético básico puede ser la siguiente:

1. Inicialice una población de cromosomas
2. Decodifique los cromosomas en individuos.
3. Evalúe el desempeño de los individuos.
4. Suprima los individuos de peor desempeño.
5. Cree nuevos cromosomas combinando cromosomas actuales. Durante este paso, aplique recombinación y mutaciones.
6. Decodifique los cromosomas en nuevos individuos.
7. Si no ha terminado el lapso de la simulación vuelva al paso 3.

Este algoritmo es sólo un núcleo básico que se puede variar en ciertos detalles al implementarlo. Formalizando un poco este procedimiento, tenemos :

```

procedure AG
begin
    inicialice el conjunto C de cromosomas

    while not (condición de terminación) do
    begin
        obtenga el conjunto I de individuos:      I = d(C)
        evalúe desempeño de todos los individuos: D = f(I)
        suprima los cromosomas de peores individuos: C = C - P
        cree nuevos cromosomas:                  N = recombinación(C)
        permita mutaciones:                      N = mutación(N)
        actualice conjunto de cromosomas:       C = C + N
    end
end

```

En la práctica muchas veces los cromosomas de los conjuntos C y N se representan como sargas (strings), a menudo binarias de K bits, por ejemplo, un cromosoma puede ser, con K=19 :

1001010111011101011

en el cual, por convención acordamos qué característica del individuo está codificada en cada bit o subsarta de bits. La función de decodificación d lo que hace es obtener un "objeto" al cual le podamos evaluar un valor de "desempeño", o cifra de mérito. Ese "objeto" hace el papel de "individuo" para nuestros propósitos. El subconjunto P está formado por aquellos miembros de I que tienen un desempeño inferior a cierto valor umbral, o podemos hacer que P sea una fracción determinada de individuos (ejemplo: un diez por ciento de I).

La operación de mutación puede ser implementada por medio de variaciones al azar (random), y para la operación de recombinación existen muchas alternativas, una utilizada a menudo es obtener un cromosoma "hijo" tomando los primeros J bits de la izquierda del cromosoma "padre" y concatenándole los $K-J$ bits de la derecha del cromosoma "madre".

Las propiedades adaptativas de los algoritmos genéticos se han utilizado con buenos resultados para resolver, por ejemplo, problemas de diseño óptimo.

11. TENDENCIAS ACTUALES DE LA INVESTIGACION

El diseño de buenos algoritmos de aprendizaje tiene hasta el presente mucho de "arte", muchos de los investigadores opinan que el campo debe madurar buscando bases teóricas adecuadas que permitan construir una "ciencia" sobre el aprendizaje.

Considerando un método de aprendizaje como un "algoritmo", se le pueden aplicar la técnica y la perspectiva de la Teoría general de algoritmos y complejidad computacional, las cuales podrían suministrar bases teóricas sólidas para la ciencia del aprendizaje.

Los primeros esfuerzos en este sentido fueron hechos por Gold en 1967 (trabajo comentado por Angluin y Smith en 1983). En 1984 Valiant introduce un esquema probabilístico para el estudio de algoritmos de aprendizaje. En 1987 han estudiado las aplicaciones de este esquema investigadores como : Kearns, Li, Pitt , Valiant , Haussler y Rivest.

El esquema de Valiant ha sido más exitoso que el de Gold al llevar a la práctica un algoritmo de aprendizaje. Para permitir mayor eficiencia requiere sólo que se encuentre una buena aproximación al concepto objetivo sin requerir una identificación exacta de ese concepto. Se ha demostrado que existen algoritmos eficientes para conseguir este tipo de identificación para varias clases de conceptos objetivo. Sin embargo, ese esquema tiene deficiencias en los siguientes factores :

- El problema de diseñar algoritmos incrementales de aprendizaje que sean eficientes.
- La relación entre el desempeño probabilístico y el del "caso peor"
- El papel de las consultas generadas por el sistema discente, en la identificación del concepto
- El problema del ruido en los ejemplos de entrenamiento, y
- El problema del aprendizaje no supervisado

En general, el campo del aprendizaje de la máquina aparece actualmente como sumamente diverso en cuanto a métodos y terminología. Tal vez en el futuro se logre un mayor grado de coherencia entre enfoques tan diferentes.

En cuanto a las Redes Neuronales Artificiales, la mayoría de los modelos usuales permiten a estas ajustar sus comportamientos cambiando los pesos de interconexión entre neuronas, pero el número de neuronas, y las relaciones estructurales o arquitectura de la red deben ser establecidas de antemano por el diseñador del sistema. Una vez establecida la estructura, esta permanece fija durante el ciclo de vida de la red. Actualmente se investiga la adaptación a nivel estructural, o

sea permitiendo cambios en la arquitectura, como parte del proceso adaptativo (véase Lee, T-Ch.). Para implementar estos procedimientos algunos investigadores, como S.A. Harp y T. Samad, han aplicado las metodologías de los algoritmos genéticos para ir llegando a arquitecturas mejora adaptadas.

BIBLIOGRAFIA RECOMENDADA

- ANGLUIN, D. y C. SMITH **Inductive Inference : theory and methods**, en revista *Computing Surveys*, 15, 237- 269; 1983
- BOOSE, J. H. : **A Survey of Knowledge Acquisition Techniques and Tools**, en revista *Knowledge Acquisition* , 1(1):3, 1988
- BOOSE, J. H. y Gaines, B. (editores) : **Knowledge Acquisition Tools for Expert Systems**, Academic Press, Londres, 1988
- BREIMAN, L., FRIEDMAN, J. et al. : **Classification and Regression Trees**, Wadsworth International, Belmont, California, 1984
- GALLANT, S : **Connectionist Expert Systems**, en revista *Communications of the ACM*, febrero 1988
- LEE, T-Ch. **Structure Level Adaptation for Artificial Neural Networks**, Kluwer Academic Publishers, 1991
- MICHALSKI, R.S., J.G. CARBONELL y T.M. MITCHELL, editores: **Machine Learning, An artificial Intelligence Approach, Vol. I** , Tioga Publishing Company, 1983
- MICHALSKI, R.S., J.G. CARBONELL y T.M. MITCHELL, editores: **Machine Learning, An artificial Intelligence Approach, Vol. II**, Morgan Kaufmann Publishers inc., 1986.
- MICHALSKI, R.S., J.G. CARBONELL y T.M. MITCHELL, editores: **Machine Learning, A Guide to Current Research**, Kluwer Academic Publishers, 1986.
- NARENDRA, K. y THATHACHAR, M.A.L. : **Learning Automata**, Prentice-Hall, 1989.
- NATARAJAN, B. K. **Machine Learning, a theoretical approach**, Morgan Kaufmann Publishers Inc., 1991
- QUINLAN, J. R. : **Knowledge Acquisition from Structured Data**, en revista *IEEE EXPERT*, dic. 1991, pp. 32-37.
- SAMPSON, J. R. **Adaptive Information Processing, An introductory survey**, Springer Verlag, 1976
- SASTRY, S. y BODSON, M.: **Adaptive Control**, Prentice-Hall, 1989
- VALIANT, L. G.: **A Theory of the Learnable**, en revista *Communications of the ACM*, Vol. 27, N° 11, pp. 1134-1142, Association For Computing Machinery.