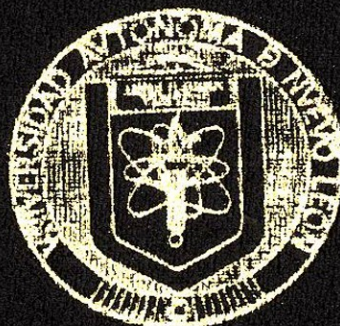


UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO MATEMATICAS



TECNICAS DE APRENDIZAJE MAQUINAL

TESIS

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES**

PRESENTA

MARIA SOLEDAD MARTINEZ SOLIS

SAN NICOLAS DE LOS GARZA, N. L.

JULIO DE 1990

TL

Q325

.5

M37

1990

c.1



1080171489

08/11/90

Presidente: Ing. Aurelio Ramirez G.

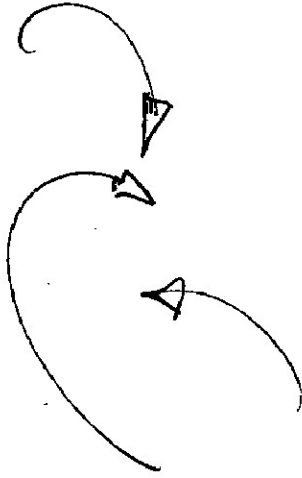
Sec. : Lic. Leonardo F. Salazar -

Tec. : Lic. O. Aguilar

7:10

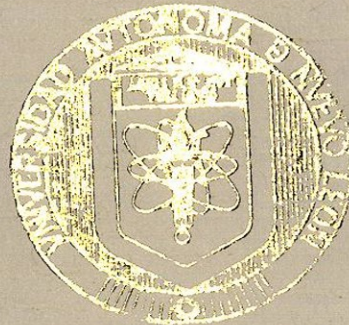
1 hora

8:20



UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO MATEMATICAS



TECNICAS DE APRENDIZAJE MAQUINAL

TESIS

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES

PRESENTA

MARIA SOLEDAD MARTINEZ SOLIS

SAN NICOLAS DE LOS GARZA, N. L.

JULIO DE 1990





TECNICAS DE
APRENDIZAJE MAQUINAL

C O N T E N I D O

I. INTRODUCCION	1
- Breve historia de la Inteligencia Artificial	
- Breve historia del aprendizaje maquina.	
II. PROLOGO.	4
III. DEFINICION DEL OBJETIVO.	6
IV. LENGUAJES DECLARATIVOS.	7
V. PROGRAMA INTELIGENTE.	19
VI. LOCALIZACION DEL APRENDIZAJE.	23
VII. CLASES DE APRENDIZAJE	27
- Aprendizaje de memoria	
- Aprendizaje cognoscitivo.	
VIII. ADQUISICION DEL CONOCIMIENTO.	33
IX. APROXIMACION A LAS REDES NEURALES	39
- La regla básica delta	
- Limitaciones de la regla básica delta	
- Propagación hacia atrás.	
X. SELECCION DE TECNICAS.	53
XI. CONCLUSIONES.	54

XII. APENDICE A	57
Ejemplo de redes neurales.	
XIII. APENDICE B	64
Ejemplo de PROLOG.	
XIV. APENDICE C	67
Glosario de terminología del ambiente.	
XV. REFERENCIAS.	84

A mamá

Ma. E. del Socorro Solís G.

*Por su comprensión y apoyo,
pero principalmente por ser
una gran mujer.*

*A la memoria de papá
José G. Martínez O.
Por los recuerdos y
enseñanzas que me
dejó.*

*A mi hermana Mariha
Por el apoyo que
siempre me brindó.*

*A mi hermana Lorena
Por su ayuda desinteresada.*

*A mi hermana Lupita
Por ser la consentida.*

Quiero expresar mi agradecimiento a todas las personas que de alguna manera contribuyeron a la realización de este trabajo, al Ing. Antonio López por las facilidades que me brindó permitiéndome el uso del equipo y el software disponibles, al Ing. Aurelio Ramírez por su asesoría, al Lic. Jorge Loredó Gerente del Depto. de Cerámica y Materiales de Vitro Tec y un agradecimiento muy especial al Ing. Arturo Ortega por motivarme en todo momento para que terminara este trabajo lo antes posible.

Ma. Soledad Mtz. S.

I. INTRODUCCION

BREVE HISTORIA DE LA INTELIGENCIA ARTIFICIAL (IA).

Es difícil establecer una fecha del inicio de la IA. Quizá el crédito por el nacimiento debería darse a A. M. Turing por su invención de la computadora de programa almacenado, recuerde que las primeras computadoras fueron concretamente máquinas dedicadas que literalmente tenían que ser recableadas para resolver diferentes problemas. El invento de Turing hizo posible cambiar la función de la computadora rápida y fácilmente mediante la ejecución de un nuevo programa.

Sin embargo, lo que se conoce comúnmente como IA comenzó alrededor de 1960 cuando John McCarthy creó el primer lenguaje de IA el LISP.

Este término se imputa generalmente a Marvin Minsky quien en 1961 escribió un artículo titulado "Pasos hacia la Inteligencia Artificial (Enero 1961)". Los años sesentas fueron una época de intenso optimismo sobre la posibilidad de hacer pensar a las computadoras. Después de todo, fue la década en ver a la primera computadora que jugó ajedrez, las primeras demostraciones matemáticas por computadora y el famoso programa ELIZA, este programa fue escrito por Joseph Weizenbaum, y actuaba como un psicoanalista Rogeriano. En este tipo de análisis, el psiquiatra toma un papel pasivo, generalmente repitiendo simplemente las propias respuestas del paciente más que dirigiendo la conversación -justo el tipo de tarea que puede hacer fácilmente una computadora-.

En esta época parecía que el objetivo de producir un programa que tuviera inteligencia como la humana estaba a la vuelta de la esquina, sin embargo un problema común era que los programadores intentaban incrementar la generalidad de ciertos

programas, estos intentos requerían mayores recursos informáticos de los que existían. Por ejemplo, o la memoria se acababa rápidamente o el tiempo de ejecución se convertía en demasiado lento.

Por la mitad de los setentas, fueron comunes las computadoras con grandes memorias y las velocidades de cálculo se habían incrementado espectacularmente. Aún así, muchos de los antiguos enfoques de IA seguían fallando debido a que había ineficiencias inherentes. A finales de los setentas se alcanzaron varios éxitos tales como procesamiento de lenguaje natural, representación del conocimiento y resolución de problemas en áreas específicas de la IA.

BREVE HISTORIA DEL APRENDIZAJE MAQUINAL.

La aproximación al aprendizaje desarrollado por la Inteligencia Artificial, es una disciplina científica muy nueva, cuyo origen puede ser localizado a mediados de los setentas y su primera manifestación está constituida por los documentos de "First Machine Learning Workshop", los cuales tuvieron lugar en la Universidad de Carnegie-Mellon.

El primer intento de aprendizaje por computadoras se remonta 25 años atrás aproximadamente. El cual consiste principalmente de un intento para el modelo de reconocimiento de imágenes. Sus características comunes están en que intentan describir un sistema "creciente" donde el conocimiento es casi nulo al empezar pero crece progresivamente durante la experimentación del sistema.

El más famoso de estos modelos es el perceptron de F. Rosenblatt [Rosenblatt 1958], cuyas limitaciones son mostradas por Minsky y Pappert [Minsky & Pappert 1969]. Sin embargo, estas limitaciones han sido rechazadas recientemente por la nueva aproximación conexionista [Touretzky & Hinton 1985].

El resultado más espectacular obtenido en este período fue de Samuel (1959,1963). Consiste de un sistema el cual aprende a jugar damas, y alcanza el dominio completo del aprendizaje. Un estado detallado de este programa nos permite entender como desengaña las posibilidades fantásticas surgidas después de este suceso. En efecto, Samuel ha provisto su programa con una serie de parámetros, cada uno de los cuales fue capaz de tomar valores numéricos. Estos valores numéricos fueron ajustados por su experiencia e ingenio. Realmente todo el conocimiento fue incluido en la definición de los parámetros, antes que en los valores numéricos asociados.

Durante los sesentas surgió otra aproximación: de aprendizaje simbólico, orientada hacia la adquisición de conceptos y conocimiento estructurado. El más famoso de los soportes de esta aproximación es Winston (1975) y el resultado más espectacular fue obtenido por el programa de Buchanan META-DENTRAL [Buchanan 1978] el cual genera reglas que explican los datos de la espectroscopia de la masa usados por el sistema experto DENTRAL [Buchanan 1971].

Como se escribió anteriormente, comenzó una nueva aproximación que consiste en el reconocimiento de los sucesos principales del pasado, una masa importante del conocimiento fue usada en sus sistemas implícitamente. Cómo puede estar incluida explícitamente? Y antes como todo pudo ser controlado, aumentado y modificado? Estos problemas parecen importantes a una porción cada vez más grande de investigadores de Inteligencia Artificial (IA).

II. PROLOGO

Este trabajo presenta las diferentes técnicas de aprendizaje maquina existentes empezando desde las técnicas básicas que son el aprendizaje de memoria y el cognoscitivo hasta la numérica, conceptual, premio/castigo, propagación hacia atrás, etc., se dedica además un capítulo a la tecnología de las redes neurales, no sin antes describir la teoría básica que permita entender más claramente los conceptos que se manejan, de esta manera al inicio del trabajo se da una definición de lo que son los lenguajes declarativos y un programa inteligente, además se dedica otro capítulo a la localización del aprendizaje.

La razón por la cual se desarrolló este tema es por el gran auge que ha tenido la Inteligencia Artificial en los últimos años y principalmente por la inquietud de conocer un tema abstracto, interesante, que ha logrado grandes avances y sobre todo que se encuentra en pleno desarrollo.

El primer tema de interés que se aborda es el de los lenguajes declarativos, aquí se presenta la diferencia entre estos y los procedurales, este capítulo da la pauta para conocer lo que es un programa inteligente y por tal razón se presenta una discusión que permite encontrar la definición a este concepto. Posteriormente, una vez que se tienen en mente los conceptos básicos entramos a discutir el tema de la localización del aprendizaje, en este capítulo se presenta un ejemplo sencillo que permite un entendimiento más claro del tema, una vez que ya sabemos donde se localiza el aprendizaje es necesario conocer las técnicas básicas del aprendizaje y posteriormente las que se

derivan de estas que son vistas en el capítulo de adquisición del conocimiento. Enseguida se dedica un capítulo a la tecnología de las redes neurales por considerar que representan un tema muy interesante dado que intentan representar el conocimiento de la misma manera que se representa en el cerebro humano. En un capítulo adicional se intenta seleccionar a la mejor o las mejores técnicas de aprendizaje maquina existentes.

Monterrey, N.L. ; a 2 de Julio de 1990

Ma. Soledad Martínez S.

III. DEFINICION DEL OBJETIVO

Uno de los objetivos es aprender las diferencias que existen entre los lenguajes procedurales y los declarativos, además de entender que es un programa inteligente y, así como en los humanos el aprendizaje se localiza en algún lugar del cerebro, en un programa inteligente donde se localiza.

Pero el objetivo principal está en mostrar las técnicas de aprendizaje maquinales más importantes que se conocen actualmente empezando desde la más primitiva (aprendizaje de memoria) hasta la más compleja y sofisticada (aprendizaje inventivo), además de mostrar algunos ejemplos sencillos utilizando los paquetes que se tienen disponibles (PROLOG y NEUROHELL) para hacer más entendible el concepto de aprendizaje maquinales. Al final de este trabajo se espera dar a conocer cual es la mejor técnica de aprendizaje maquinales, si es que la hay.

IV. LENGUAJES DECLARATIVOS

Podemos hacer una distinción entre los lenguajes declarativos y los procedurales. Los lenguajes procedurales son aquellos que conocemos como lenguajes de alto nivel (COBOL, PASCAL, FORTRAN, BASIC, etc.), este tipo de lenguajes tienen una serie de estructuras ya definidas, como lo son: IF/THEN, WHILE/DO, etc. Con este tipo de lenguajes es posible desarrollar todo tipo de aplicaciones, sin embargo, si se va a desarrollar un Sistema Experto la aplicación se complicaría demasiado si utilizamos este tipo de lenguaje, debido a que este tipo de sistemas requieren instrucciones de un nivel más elevado que le permitan realizar su función de una manera más rápida utilizando un menor número de instrucciones. Los lenguajes declarativos están desarrollados sobre esta base, se hicieron precisamente para realizar aplicaciones en el área de la Inteligencia Artificial. El primer lenguaje declarativo se desarrolló por la década de los 60's y se le conoce como LISP creado por John McCarthy, a este le siguieron otros desarrollos del mismo tipo entre los cuales podemos mencionar el PROLOG desarrollado por Alain Colmerauer y sus colegas de la Universidad de Marsella alrededor de 1970, y es de este último del cual presentamos un ejemplo que ilustra sus instrucciones básicas y permite hacer una distinción entre los lenguajes procedurales y los declarativos. Además, utilizando el mismo ejemplo se muestra el código en PASCAL.

* Programa ejemplo */

domains

h,x = symbol

predicates

mama(h,h)
papa(h,h)
hermano(h,h)
hermana(h,h)
sexo(h,x)
tia(h,h)

clauses

mama(maria,laura).
mama(maria,julio).
mama(lucia,pedro).
mama(rosa,maria).
mama(rosa,lucia).
papa(jose,juan).
papa(jose,laura).
sexo(laura,f).
sexo(maria,f).
sexo(lucia,f).
sexo(julio,m).
sexo(pedro,m).
sexo(juan,m).

hermano(X,Y) if
mama(Z,X) and mama(Z,Y) and
sexo(X,m) or
papa(Z,X) and papa(Z,Y) and
sexo(X,m).

hermana(X,Y) if
mama(Z,X) and mama(Z,Y) and
sexo(X,f) or
papa(Z,X) and papa(Z,Y) and
sexo(X,f).

tia(X,Y) if
mama(Z,Y) and hermana(X,Z) or
papa(Z,Y) and hermana(X,Z).

DESCRIPCION DEL EJEMPLO:

Este ejemplo muestra como con un menor número de instrucciones PROLOG podemos desarrollar aplicaciones que en un lenguaje procedural requeriría de más líneas de código. La idea básica es la de desarrollar un programa que tome como entrada una base de datos que contiene los siguientes tipos de parejas: (madre,hijo), (padre,hijo), (persona,sexo); y en base a esta información el programa debe ser capaz de responder preguntas como: Quién es el hermano de una persona X?, Quién es la hermana de una persona X? y Quién es la tía de una persona X?, además de responder a otras más sencillas como: Quién es la mamá de una persona X?, Quién es el papá de una persona X? yCuál es el sexo de una persona X?. Estas tres últimas preguntas pueden ser consultadas directamente de la base de datos, pero las tres primeras requieren de una búsqueda que permita relacionar los registros de la base de datos entre sí en base a una serie de condiciones que conduzcan a su solución.

Si pensamos en desarrollar la idea en un lenguaje procedural, PASCAL por ejemplo, inmediatamente imaginamos que esto se haría utilizando una serie de instrucciones condicionales IF/THEN anidadas, lo cual se llevaría una gran cantidad de líneas de código que si utilizamos un lenguaje declarativo el cual está desarrollado precisamente sobre esta base.

El código en PROLOG nos muestra esta diferencia.

El PROLOG tiene cuatro secciones básicas que son: DOMAINS, PREDICATES, CLAUSES y GOAL.

En la sección DOMAINS se define el tipo de las variables a usar, para este ejemplo sólo se usa el tipo SYMBOL que define a variables que tomarán valores tipo caracter.

La sección PREDICATES se utiliza para declarar los predicados que serán utilizados en la siguiente sección. En el ejemplo se declaran los siguientes:

PREDICADO	SIGNIFICADO
mama(h,h)	h es mamá de otra persona h
papa(h,h)	h es papá de otra persona h
hermano(h,h)	h es hermano de otra persona h
hermana(h,h)	h es hermana de otra persona h
tia(h,h)	h es tía de otra persona h
sexo(h,x)	persona h tiene un sexo x.

En la sección de CLAUSES es donde se localiza la base de datos y donde se declara el mecanismo que define a los predicados que están en función de otros. En el ejemplo se definen tres mecanismos que son:

- hermano(X,Y), la definición está dada de la siguiente manera:
X es hermano de Y si existe una persona Z que es mamá de X y también de Y y además si X es de sexo masculino(m), o bien, si existe una persona Z que es papá de X y también de Y y además el sexo de X es masculino(m).
- hermana(X,Y), se define de la siguiente manera:
X es hermana de Y si existe una persona Z que es mamá de X y también de Y y además si X es de sexo femenino(f), o bien, si existe una persona Z que es papá de X y de Y y además X es de sexo femenino(f).

- $tia(X,Y)$, esta relación está definida como:

X es tía de Y si existe una persona Z que es mamá de Y y además X es hermana de Z.

Al correr el programa este nos responderá con un GOAL, está en espera de que le hagamos una pregunta. En la hoja siguiente se muestran los resultados de una corrida.

En el ejemplo no se utiliza la sección GOAL, pero cabe mencionar que es en ella donde se puede tomar el control de la corrida, es decir, se puede evitar que el programa responda con un GOAL al correrlo, pidiendo de otra manera la pregunta que se desea realizar. Sin embargo, el objetivo del ejemplo no es dar un curso de PROLOG sino mostrar las diferencias entre cualquier lenguaje declarativo y otro procedural.

Como se puede apreciar en esta corrida las preguntas se hacen con el mismo formato con que se declaran los predicados, y así para la primera pregunta le estamos cuestionando: Quién es el hermano de Laura?, en la siguiente le preguntamos Quiénes son los hijos de María?, y así sucesivamente.

En el código en PASCAL que se muestra en la siguiente página sólo se define la relación hermano (subrutina hermano), la cual nos permite visualizar la diferencia de líneas que existe entre este código y el de PROLOG.

Se anexa además una corrida del programa.

Program Familia;

Type
p=array [1..15,1..15] of string[15];

Var

mh,ph,ts,s: p;
X,Y,mama,papa,tia,sexo: string[15];
her: boolean;
i,j: integer;

Procedure asigna;

Begin

{ La columna 1 contiene a las mamas }
{ y de la 2 en adelante son los hijos }

mh[1,1] := 'Maria'; mh[1,2] := 'Laura'; mh[1,3] := 'Julio';
mh[2,1] := 'Lucia'; mh[2,2] := 'Pedro';
mh[3,1] := 'Rosa'; mh[3,2] := 'Maria'; mh[3,3] := 'Lucia';

{ La columna 1 contiene a los papas }
{ y de la 2 en adelante son los hijos }

ph[1,1] := 'Jose'; ph[1,2] := 'Juan'; ph[1,3] := 'Laura';

{ La columna 1 contiene el nombre de la persona }
{ y la 2 su sexo }

s[1,1] := 'Laura'; s[1,2] := 'f';
s[2,1] := 'Maria'; s[2,2] := 'f';
s[3,1] := 'Lucia'; s[3,2] := 'f';
s[4,1] := 'Julio'; s[4,2] := 'm';
s[5,1] := 'Pedro'; s[5,2] := 'm';
s[6,1] := 'Juan'; s[6,2] := 'm';

End;

Procedure hermano;

Var

swm1,swm2,swp1,swp2,sws: integer;

Begin

swm1:=0; swm2:=0; swp1:=0; swp2:=0; sws:=0;

{ Buscando el sexo de X }

For i:=1 to 6 do

Begin

 If s[i,1]=X then

 Begin

 If s[i,2]='m' then sws := 1;

 end;

end;

{ Buscando a mama }

```

For i:=1 to 3 do
Begin
  For j:=2 to 3 do
  Begin
    If X = mh[i,j] then
      Begin
        mama := mh[i,1];
        swm1 := 1;
      end;
    end;
  end;
end;
For i:=1 to 3 do
Begin
  For j:=2 to 3 do
  Begin
    If Y = mh[i,j] then
      Begin
        If mh[i,1] = mama then swm2 := 1;
      end;
    end;
  end;
end;

{ Buscando a papa }

For i:=1 to 1 do
Begin
  For j:=2 to 3 do
  Begin
    If X = ph[i,j] then
      Begin
        papa := ph[i,1];
        swp1 := 1;
      end;
    end;
  end;
end;
For i:=1 to 1 do
Begin
  For j:=2 to 3 do
  Begin
    If Y = ph[i,j] then
      Begin
        If ph[i,1] = papa then swp2 := 1;
      end;
    end;
  end;
end;

If ((swm1=1) and (swm2=1) and (sws=1))
  or ((swp1=1) and (swp2=1) and (sws=1)) then
her:=true
else
her:=false;

end;

```

{ Programa Principal }

BEGIN

```
clrscr;  
gotoxy(32,8);  
write('Relacion hermano');  
gotoxy(23,10);  
write('hermano(X,Y).- X es hermano de Y ?');  
gotoxy(32,13);  
write('X ? ');  
read(X);  
gotoxy(32,14);  
write('Y ? ');  
read(Y);  
Asigna;  
Hermano;  
gotoxy(23,17);  
If her=true then  
    write('Si, ',X,' es hermano de ',Y,'.')  
else  
    write('No, ',X,' no es hermano de ',Y,'.');
```

END.

Relacion hermano

hermano(X,Y).- X es hermano de Y ?

X ? Julio

Y ? Laura

Si, Julio es hermano de Laura.

Relacion hermano

hermano(X,Y).- X es hermano de Y ?

X ? Pedro

Y ? Lucia

No, Pedro no es hermano de Lucia.

Relacion hermano

hermano(X,Y).- X es hermano de Y ?

X ? Juan

Y ? Laura

Si, Juan es hermano de Laura.

Hasta este punto ya conocemos la diferencia entre lo que es un lenguaje procedural y un lenguaje declarativo y sabemos que este último se utiliza para desarrollos de Inteligencia Artificial, es decir, para desarrollo de programas inteligentes, sin embargo, hasta este momento no tenemos una definición del concepto de programa inteligente, el siguiente capítulo nos define este concepto y nos permite formar nuestras propias ideas acerca de este tema.

V. PROGRAMA INTELIGENTE

Para iniciar este tema, definiremos primero el concepto de inteligencia, la cual según el diccionario es: *la capacidad para percibir hechos, proposiciones y sus relaciones y razonar sobre ellos.*

Si analizamos la primera parte de esta definición, percibir hechos, proposiciones y sus relaciones, nos daremos cuenta de que una computadora es capaz de realizar estas funciones. Por ejemplo, una base de datos relacional puede almacenar (percibir) información, aceptar preguntas (proposiciones) y, como su nombre lo implica, representar relaciones. También es cierto que algunos tipos de información, tales como imágenes visuales, son difíciles de percibir, pero la definición de inteligencia no exige que la percepción tome una forma específica. Por tanto, lo que las computadoras hacen normalmente satisface el primer requisito de inteligencia.

Sin embargo, Puede la base de datos razonar sobre estos hechos, que es el segundo requisito de la inteligencia? La respuesta a esta pregunta depende de la definición que se tenga de *razonar*. Si acepta la manipulación de información de la base de datos - el acto de buscar, clasificar, procesar preguntas, archivar y similares - como razonar, entonces debe decir que la base de datos es un programa inteligente. Esto implica que la mayoría de los programas de computadora son inteligentes. Recuerde que la mayoría de los programas de computadora manipulan información en forma lógica y razonable. Por tanto, esta forma de razonar debe clasificarse como inteligencia.

De algún modo, esta conclusión es difícil de aceptar. Implica que virtualmente todos los programas pertenecen al campo de la Inteligencia Artificial, lo cual no es cierto. Existe una diferencia, pero, cuál es? Se podría decir que una base de datos relacional no puede ser un programa pensante ya que lo que hace no es análogo a pensar en la forma en que la gente piensa. Pero entonces se enfrenta al hecho de que cuando la misma tarea la realiza un archivador, obviamente demanda inteligencia por su parte. Por consiguiente, surge una paradoja: si una base de datos realiza la tarea no está pensando; no obstante, si la misma tarea la realiza una persona, entonces esa persona está pensando.

Este problema permanece, en parte, por orgullo. Es difícil aceptar que alguien más, aparte del humano, cuenta con el pensamiento cognoscitivo. Por consiguiente, cuando un programador crea un programa inteligente, la tendencia es decir, "Bueno no es realmente inteligente. Sólo 'actúa' como inteligente". No decir esto sería como admitir que el monopolio sobre el pensamiento cognoscitivo había sido perdido.

Este problema puede ser visto de una forma distinta. Uno podría decir que un perro adiestrado es inteligente si lleva el periódico a su dueño desde el porche. Uno podría incluso decir que una niña o niño de un año es bastante inteligente si hace la misma tarea. Sin embargo, si es un robot quien la realiza, la mayoría de las personas dirían que el robot no es inteligente, ya que trae el periódico porque simplemente está programado para hacerlo, y que el robot no está pensando sobre la tarea, sino simplemente haciéndola.

Otra cuestión problemática es la de el libre albedrío. A través de la historia, el pensamiento siempre ha estado conectado con el concepto del libre albedrío. Esto es, el acto de pensar

sólo puede ser realizado por una entidad con albedrío para pensar. Lo que hace a este concepto problemático, es que en el ejemplo del robot y el periódico, aparece que la niña o el niño y el perro, ambos, eligen llevar el periódico (frente a hacer alguna otra cosa), en cambio el robot está programado para hacerlo, el no puede elegir hacer otra cosa. Sin embargo, puede una computadora elegir alguna vez hacer algo? Hay programadores que afirman que "una máquina es una máquina": no puede tener libre albedrío porque no tiene mente, sólo circuitos. Este argumento es convincente. Sin embargo, otros programadores plantean lo siguiente: imagine que una computadora está monitoreando a un camión que se carga con ladrillos, y cuando la carga llega a un límite la computadora detiene la cinta que transporta a los ladrillos. La computadora decidió parar la carga, claramente ella llevaba el control de la situación y eligió parar la carga cuando esta alcanzó cierto nivel. Si la computadora no tomó esta decisión, entonces quién lo hizo? Los que proponen este argumento establecen que la habilidad de la computadora para realizar una bifurcación condicional constituye la habilidad para tomar decisiones.

Las dificultades que se ilustran en esta discusión hacen alusión a un fallo en la definición de inteligencia. Lo que el diccionario omite es el hecho de que el término inteligencia implica inteligencia humana. Con esta implicación se hace imposible admitir que una máquina pueda pensar o que un programa ejecutándose en una computadora sea inteligente, porque la mayoría de los programas no hacen la misma tarea de la misma forma que una persona. Por otra parte, cuando se elimina esta implicación es fácil decir que todos los programas son inteligentes. Que un programa sea inteligente requiere que actúe inteligentemente, es

decir, como un humano. Así, he aquí la definición de un programa inteligente:

Un programa inteligente exhibe un comportamiento similar al de un humano cuando se enfrentan a un problema similar. No es necesario que el programa resuelva concretamente, o intente resolver, el problema de la misma forma que un humano.

Obsérvese que el programa no necesita pensar como un humano, pero debe actuar como tal. (Después de todo, incluso la gente no piensa siempre de la misma forma.)

Por consiguiente, los programas inteligentes, de alguna manera, exhiben un comportamiento inteligente análogo al humano, mientras que los programas no inteligentes no lo hacen.

Ahora ya sabemos distinguir entre un programa inteligente y otro que no lo es, sin embargo, quizá en este momento surga la pregunta de en que lugar se encuentra localizado el aprendizaje, es decir, el programa inteligente donde registra lo que aprende a través de sus experimentaciones o de donde obtiene la información que utiliza para encontrar la solución a cierto problema. El capítulo siguiente nos da una explicación sencilla y muy clara de la localización del aprendizaje.

VI. LOCALIZACION DEL APRENDIZAJE

Una de las interrogantes que surgen cuando se habla de aprendizaje maquina es: dónde se localiza el aprendizaje de un Sistema Experto(SE)? Una respuesta a esta pregunta es la que a continuación se presenta:

Para iniciar esta discusión necesitamos saber que es un SE el cual se define como un programa que imita el comportamiento de un experto humano. Usa la información que suministra el usuario para devolver una opinión sobre cierta materia. De este modo, el SE le pide que responda preguntas hasta que pueda encontrar un objeto que se adecúe a sus respuestas. El siguiente ejemplo da una visión más clara del problema, considere el siguiente diálogo entre un experto en fruta y alguien que está buscando consejo:

Experto:	Es verde?
Usuario:	No.
Experto:	Es rojo?
Usuario:	Si.
Experto:	Crece en los árboles?
Usuario:	No.
Experto:	Crece en una caña?
Usuario:	Si.
Experto:	Tienen espinas las cañas?
Usuario:	Si.
Experto:	Es una frambuesa.

El objetivo de un SE en fruta es ser capaz de reproducir este diálogo. Más generalmente, un SE intenta dar consejo al usuario sobre el dominio en el que es experto.

Existen en la actualidad en el área de la Inteligencia Artificial una serie de tecnologías que se han desarrollado por las necesidades que han surgido dentro de la misma. una de estas tecnologías es la conocida como REDES NEURALES, (ver APROXIMACION A LAS REDES NEURALES), esta tecnología permite almacenar información en una estructura compuesta por nodos y conexiones. Entonces, si estamos pensando en desarrollar un Sistema Experto necesitamos saber donde se localizará el aprendizaje para de esta manera saber que unidad se accesará cuando el SE este realizando sus búsquedas. Una opción es utilizar una red neural que por su arquitectura presenta una serie de ventajas si la comparamos con otro tipo de dispositivos como por ejemplo una base de datos común.

Consideremos nuevamente el ejemplo descrito anteriormente. Hasta este momento nosotros ya conocemos que el nombre de la fruta que es de color rojo y crece en una caña que tiene espinas es "frambuesa", el SE nos arrojó este resultado después de realizar una búsqueda en su base de datos, pero, dónde se encuentra el aprendizaje del SE?, es decir, de dónde extrajo su información?, la respuesta a esta pregunta es muy fácil si recordamos que el SE debe de contar con una base de datos que contenga toda la información requerida por el problema, de esta manera podemos decir que el SE extrajo su información de su base de datos y por consiguiente su aprendizaje se encuentra localizado en ella, y si esta base de datos está representada por una red neural el aprendizaje estaría almacenado de la manera como se muestra en la figura 1.

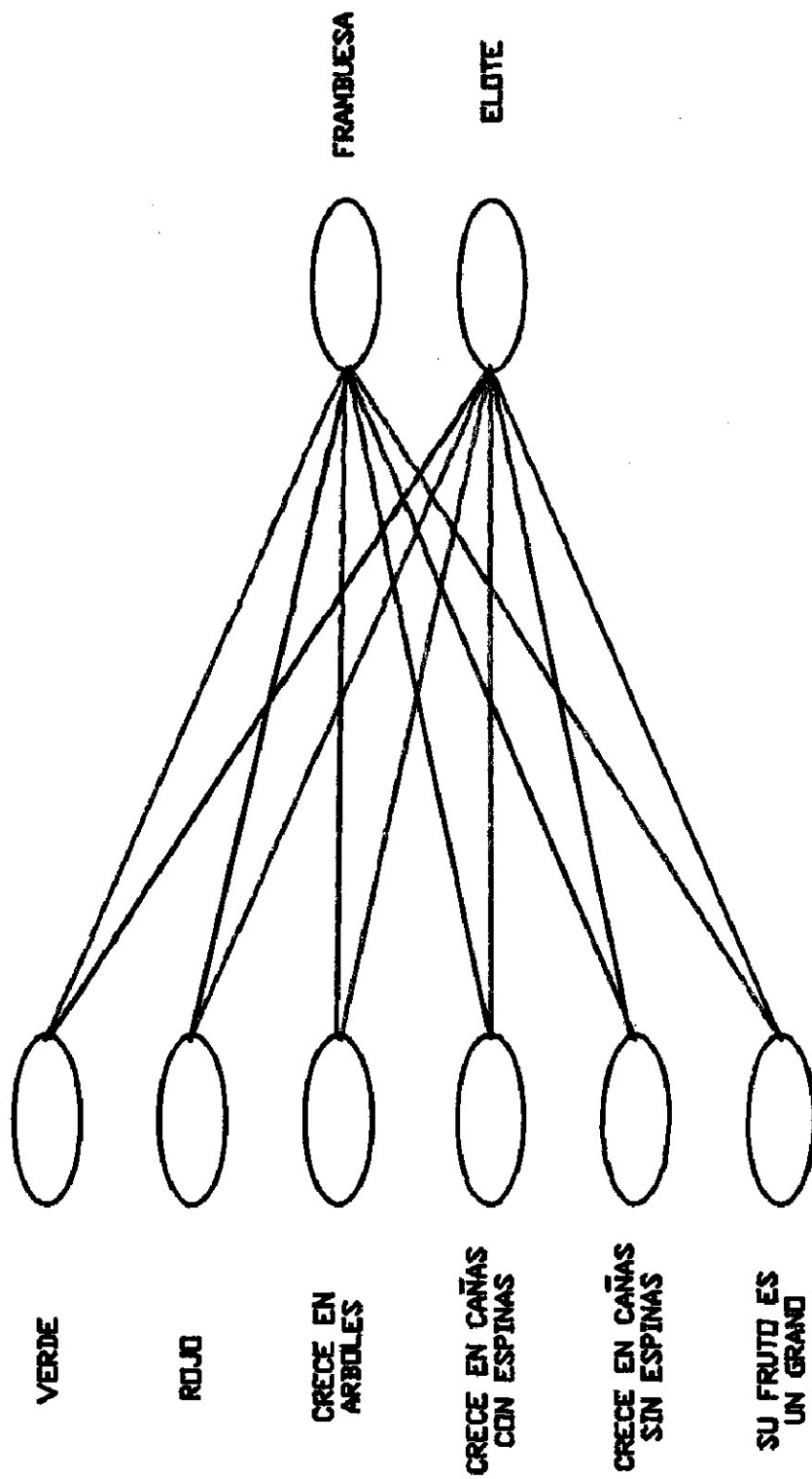


FIGURA 1
REPRESENTACION DEL APRENDIZAJE
UTILIZANDO UNA RED NEURAL

DESCRIPCION DEL EJEMPLO.

La figura 1 muestra una simple red neural donde sus nodos de entrada representan a las características del fruto y los de salida representan a los frutos, las líneas que unen a los nodos son las conexiones entre las características y soluciones, en el ejemplo las líneas rojas representan conexiones positivas y las azules conexiones negativas. Para este caso sólo se utilizan 6 nodos de entrada y 2 de salida, pero este número puede variar dependiendo de la cantidad de características y de soluciones con que cuente el problema. Así, la red reconoce dos casos que son:

- a) Si el fruto es verde, crece en cañas sin espinas y es un grano, entonces estamos hablando del elote.
- b) Si el fruto es rojo y crece en cañas con espinas, entonces es una frambuesa.

Esto será visto más detalladamente en el capítulo VIII (APROXIMACION A LAS REDES NEURALES).

Ahora ya podemos decir que el aprendizaje de un sistema se encuentra localizado en su base de datos, es ahí donde de alguna manera tiene representado su conocimiento, sin embargo, no sabemos hasta ahora como se realiza el aprendizaje, pues bien, para realizar el aprendizaje requiere de una técnica que le ayude a llevar a cabo esta tarea, y actualmente existen una serie de técnicas que están basadas en las dos que se presentan en el siguiente capítulo, el aprendizaje de memoria y el cognoscitivo.

VII. CLASES DE APRENDIZAJE

APRENDIZAJE DE MEMORIA.

CARACTERISTICAS:

- Es el más primitivo.
- Sencillo, dado que se puede representar en una simple base de datos.
- Comprende la memorización de hechos específicos como de procedimientos (llamamos procedimientos a las secuencias aprendidas de memoria).

VENTAJAS:

- Es el método más fácil de implementar en una computadora.

LIMITANTES:

- Requiere de una gran capacidad de almacenamiento.
- La computadora podrá realizar un diagnóstico después de haber analizado todos y cada uno de los hechos almacenados en la base de datos, lo cual consume mucho tiempo de proceso. A este suceso se le conoce como explosión combinatoria.

EJEMPLOS:

- Tal como se puede decir a un estudiante que el agua es líquida entre 32 y 212 grados, se puede meter la misma información a la base de datos de un programa y cuando se le pregunte cuándo es líquida el agua tanto el estudiante como la computadora contestarán correctamente.
- El método para realizar una división larga es el mismo que se utiliza para dividir dos números cualesquiera. Y así como lo aprende una persona lo podría aprender una computadora, siguiendo una serie de instrucciones programadas.

APRENDIZAJE COGNOSCITIVO.

CARACTERISTICAS:

- Parte de un concepto específico.
- En base a comparaciones, este concepto se va generalizando o restringiendo según sea el caso.
- Estas comparaciones deben hacerse en una forma suave, es decir, cada comparación entre el concepto y nuevas definiciones no deben diferir en mucho.
- Usa la razón para analizar, organizar y correlacionar piezas específicas del conocimiento, y el producto de este esfuerzo mental es la creación de descripciones de clases. Una descripción de clase es simplemente una generalización que se deriva del examen de unos pocos ejemplos concretos.

VENTAJAS:

- La computadora puede razonar en base a conocimientos adquiridos y realizar un diagnóstico que ayude a tomar decisiones posteriores.
- El almacenamiento de información es menor si lo comparamos con el requerido por el aprendizaje de memoria.

LIMITANTES:

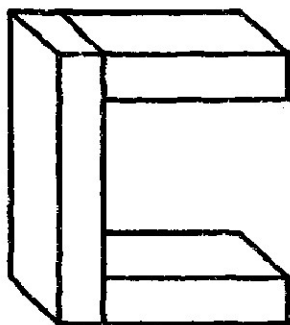
- Es difícil de implementar en una computadora.
- El equipo y software pudieran ser costosos.

EJEMPLO:

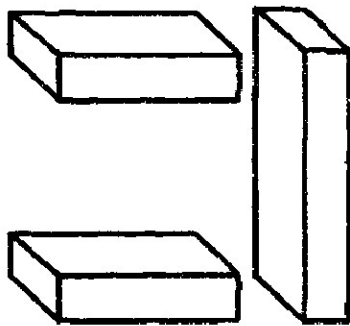
El ejemplo mostrado a continuación usa el procedimiento de acierto y fallo por poco que es un método para aprender descripciones de clases.

Imagine que usted está enseñando a alguien a reconocer un arco construido con bloques. Como primer paso usted le muestra la Figura 2a y luego le informa a la persona que ese es un arco. Luego, usted quita el bloque superior y lo coloca próximo a los otros dos Figura 2b y le dice a la persona que ese no es un arco. Esto implica que un arco debe tener un bloque en su parte superior. Enseguida, construye un arco como el mostrado en la Figura 2c e informa a la persona que ese es un arco. Entonces usted construye la estructura mostrada en la Figura 2d y le hace saber que ese no es un arco. Esto implica que los bloques de los lados no deben tocarse. Finalmente, usted construye el arco mostrado en la Figura 2e usando un cilindro en la parte superior, e indica que ese es todavía un arco. En este punto, el estudiante tendrá esta descripción de clase para un arco: debe tener o un bloque o un cilindro sobre la parte superior de otros dos bloques que no se tocan, y los bloques laterales pueden estar de pie o tumbados.

Este ejemplo muestra como se puede enseñar a un Sistema Experto las descripciones de clase de la misma manera que lo haría una persona.



2a



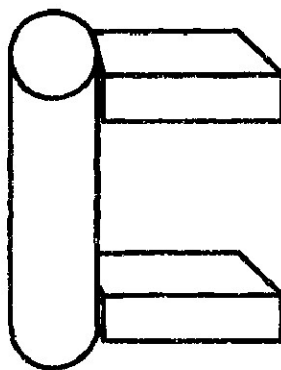
2b



2c



2d



2e

EJEMPLO TRADICIONAL
2a, UN ARCO; 2b, ARCO SIN PARTE SUPERIOR;
2c, DTRD ARCO; 2d, BLOQUES TOCANDOSE;
2e, UN ARCO

Estas técnicas de aprendizaje marcan el inicio de las que se presentan en el siguiente capítulo, representan el arranque del aprendizaje numérico, conceptual, premio/castigo, empírico, racional, deductivo, inductivo e inventivo. Yo considero que a partir del aprendizaje de memoria y el cognoscitivo se encontraron las que a continuación se describen.

VIII. ADQUISICION DEL CONOCIMIENTO

Existen tres problemas principales para la adquisición automática del conocimiento:

- El primero es de agrupamiento (el cual es llamado "clasification" en análisis de datos), y podría ser planteado de la siguiente manera: dada una masa de conocimientos, cómo pueden las características comunes de estos ser señaladas de tal modo que podamos formar sub-grupos, los que sean simples y tengan un significado? La inmensa mayoría de procedimientos por agrupamiento son numéricos por naturaleza.
- El segundo problema (de discriminación) es el de aprendizaje de procedimientos de clasificación.
Dado un conjunto de ejemplos de conceptos, cómo sería un método el cual permita cada uno de los conceptos a ser reconocidos del modo más eficiente? La gran mayoría de los métodos existentes se apoyan en evaluaciones numéricas limitadas con la disminución de una medida de entropía despues de la aplicación de descriptores.
- El tercer problema es de generalización. Empezando desde ejemplos concretos de una situación o una regla, cómo puede ser deducida una fórmula la cual sea lo suficientemente general para describir esta situación?, y cómo se puede explicar que la fórmula tiene esta capacidad descriptiva? Por ejemplo, puede ser solicitado como empezar desde una declaración equivalente a:
"Francia compra video-recorders desde Japón", la regla más general se deriva como:
"Las ciudades que no tienen suficiente desarrollo en la investigación de física del estado sólido compran equipos electrónicos en las ciudades donde los hay".

Son a estos problemas a los que se enfrentan las técnicas de aprendizaje que a continuación se describen:

SBL vs. EBL

Durante 1985 fue definida una distinción entre el Aprendizaje Basado en Semejanzas (SBL) y Aprendizaje Basado en Explicaciones (EBL).

SBL es un aprendizaje que detecta primeramente las similitudes entre un conjunto de ejemplos positivos, y en segundo término detecta las diferencias entre ejemplos positivos y negativos.

En EBL, la entrada al aprendizaje consiste de explicaciones derivadas del análisis de un ejemplo positivo o negativo, del concepto o regla que está siendo aprendido.

Generalmente, esta clase de aprendizaje está hecha con un sistema problem-solving. En cada momento, el sistema llega a una solución, ya sea un suceso o una falla (en cuyo caso sólo habla de ejemplos negativos). Un módulo analiza las razones para este suceso o falla.

Estas razones son llamadas "explicaciones" y son usadas para perfeccionar el sistema.

APRENDIZAJE NUMERICO vs. CONCEPTUAL

La aproximación numérica pretende optimizar un parámetro global. Pretende exponer un conjunto de descriptores, los cuales son los "mejores" relativos a esta optimización. También tiene

como consecuencia la generación de "clusters" de ejemplos.

Esta optimización es eficiente y resistente al ruido, pero produce reglas o conceptos los cuales son en general incomprensibles a los humanos.

La aproximación simbólica, está bien situada en cuanto a interacción con expertos humanos, pero es muy sensitiva al ruido.

Pretende optimizar una función de reconocimiento. Esta función requiere usualmente ser complementada, con medios que deben reconocer todos los ejemplos positivos, y a ser discriminantes, con medios que rechazen todos los ejemplos negativos.

Intenta expresar una afinidad conceptual entre los ejemplos.

APRENDIZAJE POR PREMIO/CASTIGO

En esta clase de aprendizaje cada vez que se obtiene un éxito, el sistema asigna más peso a las reglas que llevan a este éxito. Cada vez que obtiene un resultado negativo, reduce el peso para el uso de las reglas que utilizó.

Esta clase de aprendizaje es muy buena, desde que hace posible obtener sistemas que son independientes de sus creadores una vez que empiezan a trabajar.

Por otra parte, se puede suponer que la definición de los conceptos o reglas, de positivo y negativo depende estrechamente del conjunto de problemas. Estos sistemas son muy difíciles para aplicar superficialmente sus campos de especialización y son muy difíciles de modificar.

APRENDIZAJE EMPIRICO vs. RACIONAL

En el aprendizaje empírico, los sistemas adquieren conocimiento de una manera local. Por ejemplo, si una nueva regla ayuda con un problema solucionándolo, esta regla es agregada a la base de conocimientos, una vez asignada no contradice a las ya existentes. Cabe mencionar que esto representa un gran problema en la depuración del problema, lo que regularmente es realizado por el Ingeniero del Conocimiento.

En el aprendizaje racional, por otra parte, la adición de la nueva regla es examinada por un módulo el cual busca conectarla con el conocimiento global alrededor del Universo en donde el sistema está situado.

Está claro que el aprendizaje racional será capaz de introducir datos dependientes del medio ambiente, mientras que el aprendizaje empírico está siendo desplazado por esta razón.

(DEDUCTIVO vs. INDUCTIVO) vs. APRENDIZAJE INVENTIVO

El aprendizaje es llamado deductivo, cuando todo el conocimiento necesario está dado explícitamente desde el inicio (del aprendizaje). De esta manera, el aprendizaje numérico puede también ser descrito como deductivo.

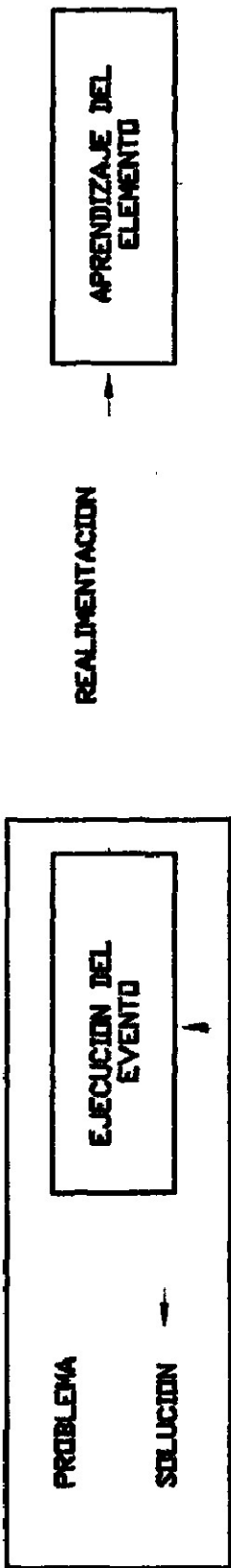
El aprendizaje inductivo, es ese en el cual aunque todo el conocimiento es dado al sistema, algo de él tiene una forma implícita. Por ejemplo, cuando el sistema inicia con una descripción inadecuada y se modifica a sí mismo, con descriptores conocidos.

El aprendizaje inventivo, es ese en el cual es necesario para el sistema crear una cierta pieza del conocimiento partiendo de nada, por ejemplo un descriptor el cual está tratando de hacer posible la definición de un concepto más útil. Este es conocido como la invención del genio, y nosotros no esperaríamos, cuanto antes, encontrar sistemas capaces de tales habilidades.

La Figura 3 ilustra como trabajaría un sistema de este tipo.

Estas técnicas de aprendizaje son utilizadas dependiendo del tipo de problema que se tenga o de las ventajas que cada una de ellas pueda presentar, sin embargo, existe una técnica más, conocida como aprendizaje de propagación hacia atrás, esta técnica es utilizada por las redes neurales y es precisamente de estas de las que hablamos en el siguiente capítulo.

ENTRADA EXTERIOR



EJECUCION DE LA INFORMACION ANADIDA



MODIFICACION DE PROGRAMA

CLASICO SISTEMA DE IA DE NO APRENDIZAJE

FIGURA 3
VISTA GENERAL DE UN SISTEMA INVENTIVO

IX. APROXIMACION A LAS REDES NEURALES

La aproximación a las redes neurales, también conocida como conexionismo o procesamiento paralelo distribuido, adopta un "cerebro metafórico" de procesamiento de información. El comportamiento inteligente en una persona parece surgir del involucramiento de las interacciones entre el gran número de neuronas.

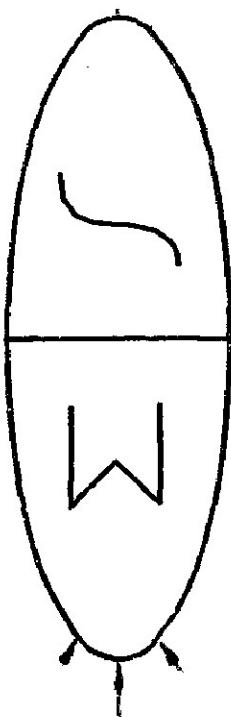
Similarmente, bajo una aproximación de red neural, el procesamiento de información ocurre hasta el involucramiento de las interacciones del gran número de neuronas simuladas, tal como lo muestra la Figura 4. Esta neurona simulada, o unidad, tiene cuatro componentes importantes:

- * CONEXIONES DE ENTRADA. Unidad que recibe la activación desde otras unidades.
- * FUNCION SUMATORIA. Combina varias activaciones de entrada en una activación singular.
- * FUNCION UMBRAL. Convierte esta sumatoria de activación de entrada en activación de salida (por ejemplo, la activación de salida pudiera ser 0 si la activación de entrada cae más abajo de algún umbral).
- * CONEXIONES DE SALIDA. Activación de salida de una unidad que llega como activación de entrada a otra unidad en el sistema.

A una conexión inter-unidad en una computadora basada en una red neural se le asigna un peso numérico que modula la activación pasando hasta la conexión. Si la conexión desde la unidad A

ENTRADAS

SALIDAS



FUNCION
SUMATORIA

FUNCION
DE UMBRAL

FIGURA 4
NEURONA SIMULADA

hasta la unidad B tiene un peso W_{BA} por ejemplo, entonces la activación de salida de la unidad A será multiplicada por este valor para determinar la activación actualmente recibida por B. Podemos entonces representar la ausencia de una conexión entre A y B para simplemente asignar un valor de cero a W_{BA} . Una relación inhibitoria entre A y B puede ser representada por un valor negativo de W_{BA} .

Este instructivo describe dos contrastes importantes entre la aproximación de la red neural y una regla más convencional basada en la aproximación encontrada en el trabajo de IA de sistemas expertos:

- * El conocimiento de una red neural se apoya en sus conexiones de inter-unidad y sus pesos. En contraste, mucho del conocimiento de un sistema experto se apoya en estas reglas (es decir, condición/acción o si/entonces).
- * Una red neural es manejada por la activación que pasa de unas unidades a otras. En contraste, un sistema experto es manejado por los símbolos generados como consecuencia de la regla-descarga.

Porque sólo numéricamente evalúa la activación que pasa de unidad a unidad en una red neural, las redes neurales son frecuentemente citadas para involucrar un nivel subsimbólico de cálculo. Una entrada, salida y estado interno de una red pueden ser caracterizadas por patrones de activación a través de sus nodos.

Cómo puede el comportamiento inteligente surgir de tales interacciones subsimbólicas de muy bajo nivel entre las unidades de una red? Esto oportunamente útil ve a cada unidad como un clasificador o detector de rasgos. Considere un ejemplo simple que involucra a Caperucita Roja, quien es enviada a la escuela para

aprender como detectar al lobo en el bosque, Caperucita Roja está observando una serie de dibujos del lobo hasta que tiene formada internamente la red mostrada en la Figura 5. Las entradas primitivas en este ejemplo son los rasgos: "orejas grandes", "ojos grandes" y "dientes grandes". Para cada rasgo, hay una unidad de entrada correspondiente con una razón de descarga (es decir, una cantidad de activación de salida) que depende del alcance al cual este rasgo se detecta en el exterior.

Estas unidades de entrada, están conectadas a una unidad correspondiente clasificada como "lobo". Las acciones de esta unidad de la función de umbral pueden causar un comportamiento como el de un AND booleano que será verdadero solamente cuando los tres rasgos del lobo sean observados. Como la unidad de umbral es baja, las funciones crecen como un OR-INCLUSIVE, tal que cualquier combinación de los rasgos del lobo es suficiente para causar la clasificación de "lobo". Es continuo el efecto combinado de un gran número de tales clasificadores y detectores de los rasgos de la unidad de entrada, este comportamiento inteligente puede surgir de una red neural.

La base numérica de la aproximación a la red neural provee un medio preparado para representar las graduaciones continuas en hechos como la intensidad de un aspecto de entrada, la certeza de una clasificación, o la importancia de una conexión entre dos unidades. En contraste, tales graduaciones están representadas con gran dificultad o no todas basadas en reglas de sistemas expertos. Las ventajas adicionales de la aproximación a la red neural pueden ir en contra de las propiedades de la asignación con fallas, continuidad de la direccionabilidad, degradación fácil, y generalización espontánea.

Pero, cómo son determinadas las conexiones y su peso en una red? Mientras el tamaño de una red se incrementa no es muy

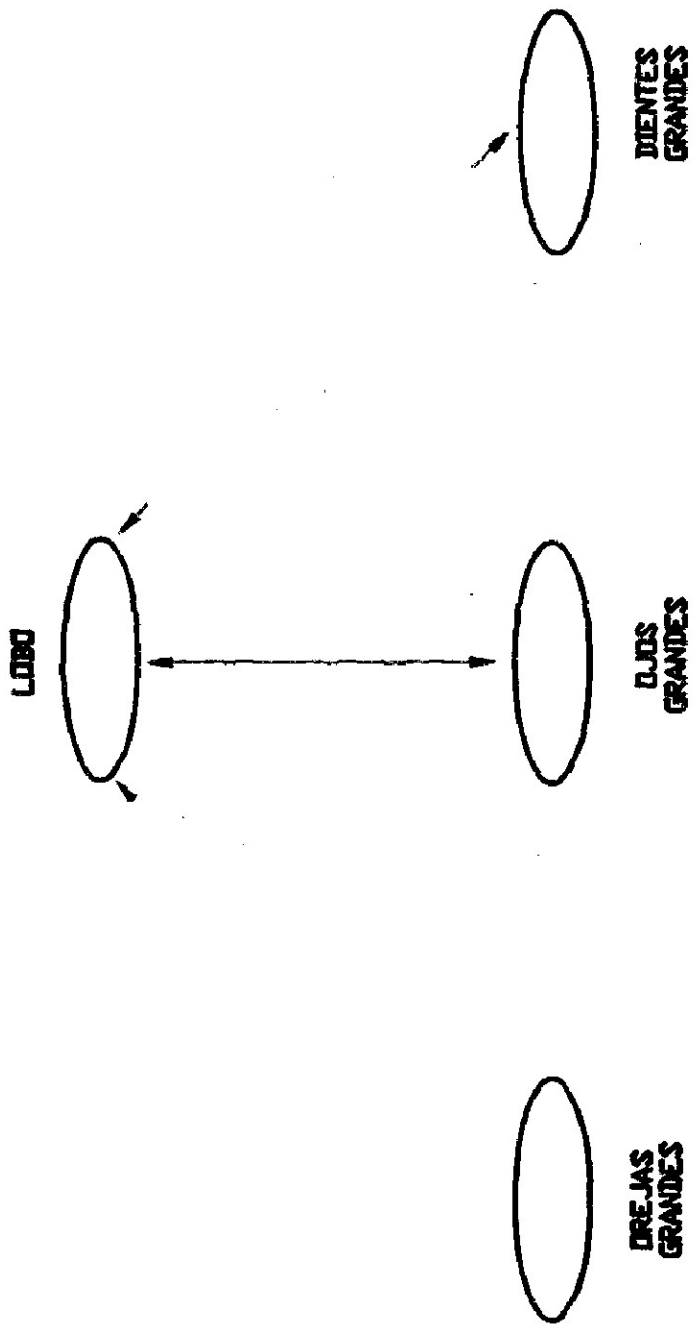


FIGURA 5
UN SIMPLE 'DETECTOR DEL LOBO'

factible para el diseñador determinar las conexiones de la red a mano, no es factible emplear la computadora en una búsqueda iterativa a fuerza bruta de las conexiones convenientes. En una red con sólo una etapa de conexiones -esto es conectando las unidades de entrada con las unidades de salida- hay un simple y elegante aprendizaje heurístico, la regla delta, esta da una red con habilidad para formar y modificar conexiones individuales en maneras que con frecuencia aproximan rápidamente a un funcionamiento óptimo. A continuación se hace una breve descripción de la regla delta, la regla más general y poderosa es la propagación hacia atrás.

LA REGLA BASICA DELTA.

Describiremos la regla delta continuando con el ejemplo de Caperucita Roja, ella encontrará a tres individuos distintos que nosotros conocemos como el lobo, la abuelita y el leñador. Limitamos a Caperucita Roja como una fase singular de conexiones entre los nodos de entrada representados por los rasgos observables y los nodos de salida representados por las acciones que Caperucita Roja pueda tomar. Ella debe aprender a escapar, gritar e indicar al leñador cuando detecte un ser con orejas grandes, ojos grandes y dientes grandes (el lobo). También debe aprender a aproximarse, besar la mejilla, y ofrecer alimento a quien sea amable, arrugado y que tenga ojos grandes (la abuelita). Y debe aprender a aproximarse, ofrecer alimento y coquetear con quien sea apuesto, amable y que tenga las orejas grandes (el leñador).

Bajo estas circunstancias, la regla delta produjo la red mostrada en la Figura 6, con las líneas rojas correspondiendo a las conexiones cargadas negativamente, y las líneas azules

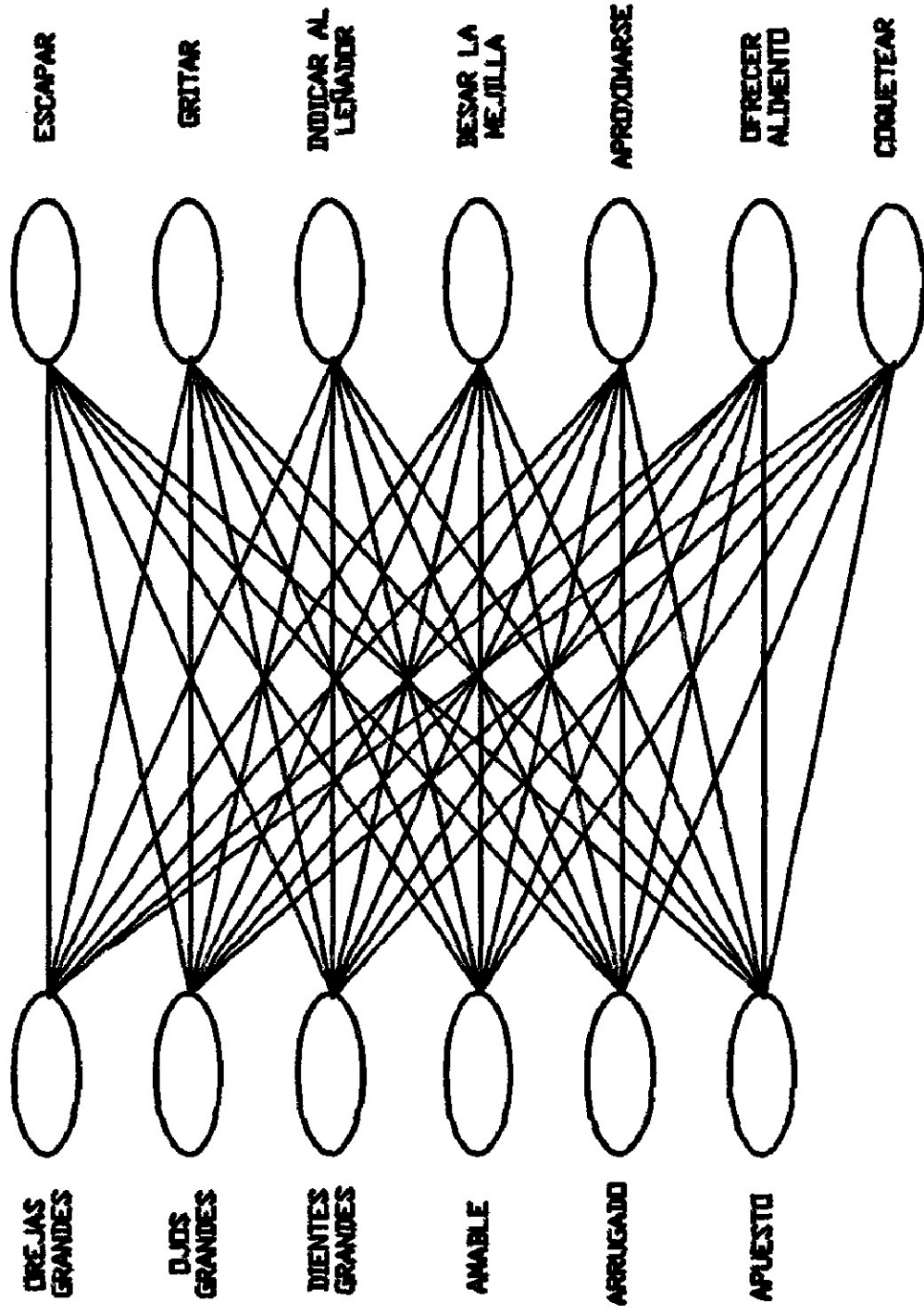


FIGURA 6
 RED NEURAL PRODUCIDA POR
 LA REGLA DE APRENDIZAJE DELTA

correspondiendo a las conexiones cargadas positivamente. Algunos aspectos en la red tienen más valor que otros (por ejemplo, "dientes grandes" vs. "ojos grandes"). Sin embargo, Caperucita Roja necesita solamente identificar correctamente dos rasgos de un individuo en el orden en que produce un conjunto apropiado de acciones usando la red de la Figura 6.

El procedimiento de enseñanza usado en conjunción con la regla delta producen integra la red de la Figura 6. Hay tres procedimientos de I/O, uno para el lobo, otro para la abuelita y otro para el leñador. Un patrón de entrada I es representado por un vector de 0s y 1s que siguen el orden de los nodos de entrada en la Figura 6. Así, el patrón de entrada para el lobo es $\langle 111000 \rangle$. El patrón de salida que queremos que Caperucita Roja produzca puede ser representado similarmente. De este modo, el patrón de salida para el lobo es $\langle 1110000 \rangle$. Antes del procedimiento los individuos, las conexiones entre todos los nodos de entrada y todos los nodos de salida son formados, y a cada conexión se le da aleatoriamente un pequeño peso inicial.

Usando el procedimiento de la regla delta se procede a formar un ciclo que se detiene hasta que un nivel satisfactorio de acciones es alcanzado. En una prueba dada, la red primero genera un patrón de salida O_p en respuesta al patrón de entrada I_p de un procedimiento par. La discrepancia, o delta, entre el actual y el anhelado comportamiento de la red es entonces determinado sustrayendo el vector O_p desde el patrón de salida T_p del patrón de enseñanza. Bajo la regla delta, el cambio de la postprueba en el peso W_{ji} de una conexión entre la unidad de entrada U_i y la unidad de salida O_j es una función de la activación I_{pi} de la unidad de entrada y el delta $(T_{pj} - O_{pj})$ asociada con la unidad de salida. Específicamente,

$$\Delta W_{ji} = n (T_{pj} - O_{pj}) I_{pi} \quad (1)$$

donde, n representa una prueba independiente de la razón de

aprendizaje. (Como la razón de aprendizaje se incrementa, así también conduce al riesgo de un comportamiento oscilatorio).

La regla delta esencialmente asigna el crédito (o la culpa) a las unidades de entrada de acuerdo a sus niveles de activación; la más activa como unidad de entrada, es la más responsable para la distribución de activación actual entre las unidades de salida. El efecto, entonces, es que esas conexiones que apuntan desde las unidades de entrada más activas de una prueba pasarán por una gran modificación en sus pesos. Al mismo tiempo, entre las conexiones que apuntan desde una unidad de entrada dada, las grandes modificaciones involucrarán a esas conexiones que señalan a las unidades de salida con grandes deltas asociados (es decir, las discrepancias entre los niveles de activación actual y el deseado).

Se puede mostrar que la regla delta pertenece a una clase de gradiente o pendiente heurística descendente. De este modo la regla delta causará una red de conexiones a cambiar en las direcciones que maximizen el cambio en un término de error que sume el cuadrado de los deltas de salida. Una manera de ver este aspecto es observar la regla delta como a un esquiador que siempre se mueve con la línea de bajada en una rápida jornada de regreso al club.

LIMITACIONES DE LA REGLA BASICA DELTA.

La regla básica delta trabaja completamente bien asignando conexiones y sus pesos en sistemas de fases singulares (involucrando solamente unidades de entrada y unidades de salida), pero es incapaz de determinar pesos a las conexiones en un sistema multifases que involucra unidades ocultas (unidades que no tienen contacto directo con el exterior). En un análisis cuidadoso hecho hace 20 años aproximadamente, Minsky y Pappert diseñaron un número de funciones interesantes que los sistemas de fase singular no pueden ejecutar. Por ejemplo, no pueden calcular el OR-EXCLUSIVE.

En un sentido, las redes de fase singular están limitadas a cálculos que mapean las vistas similares de los patrones de entrada en vistas similares de patrones de salida. En muchos cálculos tal como la relación OR-EXCLUSIVE o el cálculo más general de paridad, un pequeño cambio en el patrón de entrada puede necesitar un cambio drástico en el patrón de salida deseado. En este caso, necesitamos una red de fases múltiples con unidades ocultas. A través de las unidades ocultas, el sistema puede representar abstracciones que no pueden codificar directamente desde el medio ambiente vía los nodos de entrada.

Con el conjunto conveniente de conexiones y un gran conjunto de unidades ocultas, una red puede ejecutar cualquier mapeo entre entradas y salidas. (Es frecuente el caso de que las unidades ocultas también reducen el número de conexiones necesitadas para ejecutar un cálculo particular). Muchos de los resurgimientos actuales de interés en redes neurales puede ser trazado por el desarrollo de una generalización poderosa de la regla delta que puede llegar como un conjunto de conexiones en un sistema multifases, esta generalización es conocida como propagación hacia atrás.

PROPAGACION HACIA ATRAS.

La ecuación que determina los cambios de peso bajo la regla de propagación hacia atrás es similar en su forma a la ecuación 1. Sin embargo, la regla de propagación hacia atrás provee un medio más general del cálculo de una unidad. En una prueba dada el delta de una unidad de salida es calculada de algún modo como la regla básica delta. Específicamente,

$$\delta_{pj} = (T_{pj} - O_{pj}) f'_j(\text{net}_{pj}) \quad (2)$$

donde $f'_j(\text{net}_{pj})$ es la derivada de una función "squashing" que opera sobre la suma de las entradas de una unidad en el orden que determina la unidad de salida. Una función "squashing" es una clase especial de la función umbral que es diferenciable y no decreciente. En un sistema lineal con una función no "squashing", la salida de una unidad es igual a su entrada. En esta circunstancia especial, la delta producida por la regla de propagación hacia atrás es idéntica a la producida por la regla básica delta.

El poder real de la regla de propagación hacia atrás viene desde la asignación de deltas a las unidades ocultas que no reciben realimentación directa desde la formación de los patrones en el exterior. Esas deltas, influyen en la modificación de los pesos de las conexiones principales de las unidades ocultas. La delta para una unidad oculta es calculada como sigue:

$$\delta_{pj} = f'_j(\text{net}_{pj}) \sum_k \delta_{pk} W_{kj} \quad (3)$$

Esta es una definición recursiva en la cual el delta de la unidad está determinado por la derivada de la función "squashing" multiplicada por la sumatoria de los deltas por los pesos a la que la unidad de activación se transmite vía las conexiones de salida. Un término delta dado δ_{pk} en la sumatoria

es, de hecho, pesado por la fuerza de la conexión que apunta desde la unidad oculta U_j a la unidad U_k que es la fuente de la delta.

Como la regla de propagación hacia atrás es un nombre que sugiere, la idea básica de este cálculo de deltas para las unidades internas es propagado hacia atrás a través de los errores del sistema que están basados en las discrepancias observadas entre los valores de las unidades de salida y un patrón formado. Los deltas son primero calculados para las unidades de salida, y entonces son propagadas hacia atrás a todas las unidades que apuntan a las unidades de salida en la fase más baja. Estas unidades, a su vez, propagan sus deltas recibidos hacia atrás a las unidades que los apuntan, y así sucesivamente, hasta alcanzar el nivel de entrada. Estos deltas entonces manejan los cambios en los pesos de la red casi del mismo modo que la regla básica delta; la regla de propagación hacia atrás, como su predecesor la regla básica delta, es un gradiente descendente heurístico.

Regresamos al ejemplo de Caperucita Roja que ilustra los efectos de la regla de propagación hacia atrás con un ejemplo más concreto. Supongamos que seguiremos algún procedimiento de enseñanza para generar la red de la Figura 6, pero ahora introduciremos tres unidades ocultas. Inicialmente, cada unidad de entrada está conectada (con algún peso determinado aleatoriamente) a las tres unidades ocultas; similarmente, cada unidad oculta está conectada a todas las unidades de salida. Las conexiones no permitidas son aquellas que conectan directamente unidades de entrada con unidades de salida. Ahora tenemos dos fases de conexiones, necesitando el uso de la regla de aprendizaje de propagación hacia atrás.

La Figura 7 muestra los resultados de este experimento. Interesantemente, las unidades ocultas vienen a representar internamente los conceptos para lobos, abuelitas y leñadores. Es

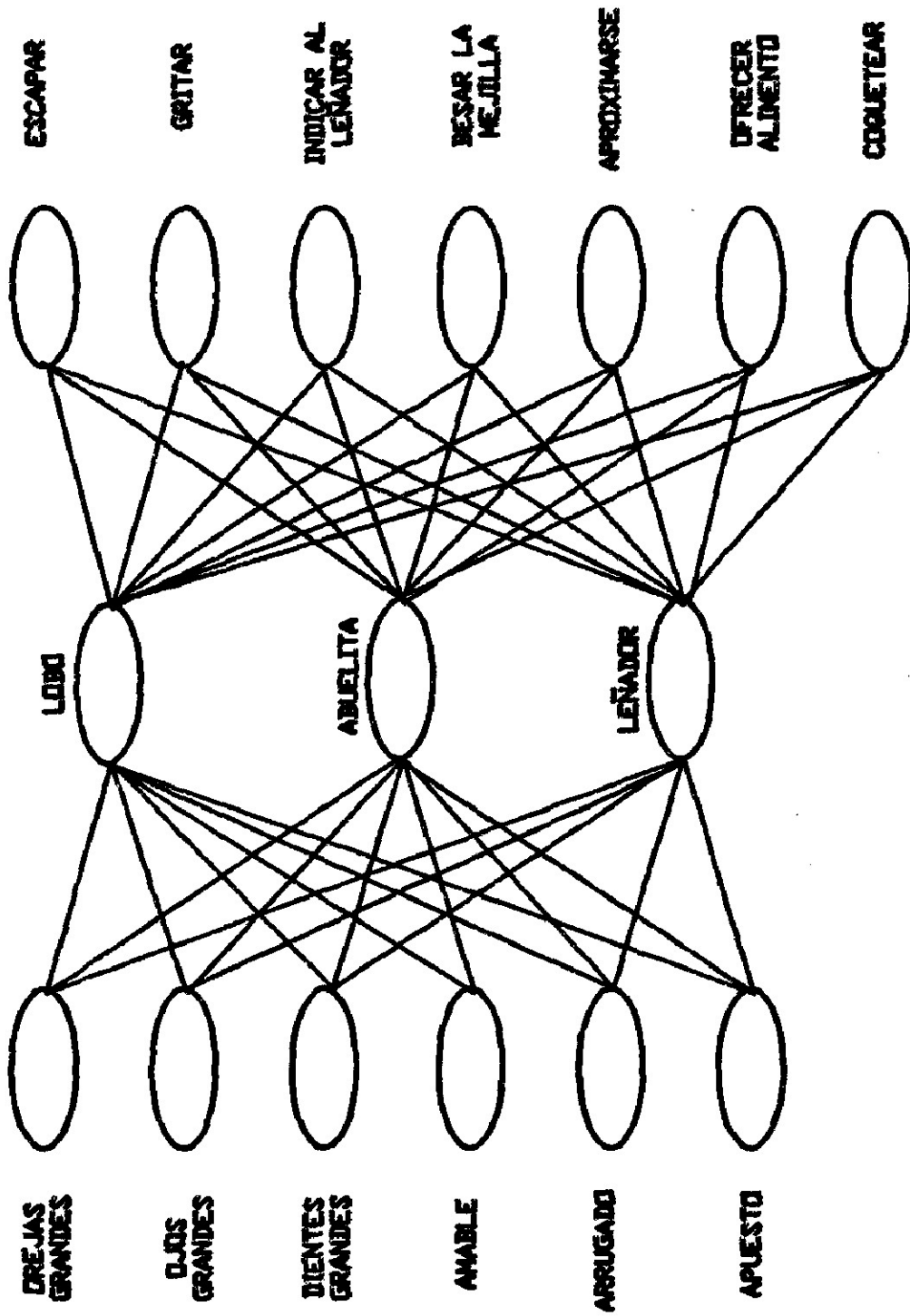


FIGURA 7
RED NEURAL PRODUCIDA POR LA REGLA
DE APRENDIZAJE DE PROPAGACION HACIA ATRAS

frecuente el caso de que las unidades ocultas, mediante las acciones de la propagación hacia atrás, vendrán a representar abstracciones muy útiles del exterior. Note que el mapeo de I/O en este ejemplo no requiere el uso de unidades ocultas. Como lo indica la Figura 6, este mapeo puede ser realizado sin las unidades ocultas usando la regla básica delta. Es, sin embargo, interesante notar que la introducción de unidades ocultas en la Figura 7 reduce el número de conexiones necesitadas para representar el mapeo.

En el apéndice A se muestra el ejemplo de Caperucita Roja utilizando el NEURO SHELL, que es un paquete de redes neurales que utiliza la regla de aprendizaje de propagación hacia atrás.

Hasta ahora ya hemos abarcado las principales técnicas de aprendizaje maquinales existentes, y después de haberlas entendido estamos preparados para definir si existe o no una técnica que es la mejor, el siguiente capítulo comprende esta definición.

X. SELECCION DE TECNICAS

Cuando se está pensando en desarrollar un SE uno de los problemas que se presentan está en elegir la mejor técnica de aprendizaje maquina existente. Sin embargo, no podemos decir que de todas las técnicas que se conocen hay una que sea la mejor sino que cada una se adapta a problemas con necesidades diferentes.

Quizá para ciertos problemas sea menos importante el tiempo de respuesta que el costo del hardware y el software requeridos por el sistema, en estos casos se optaría por elegir una técnica de aprendizaje cognoscitivo. Pero, si se prefiere sacrificar tiempo de respuesta a cambio de un SE más económico la solución es una técnica de aprendizaje de memoria.

Entonces, como se puede apreciar cada problema encontrará una técnica de aprendizaje diferente que se adapte a sus necesidades.

En la actualidad existen una serie de softwares para desarrollo de SE entre los cuales podemos mencionar al IXL (Programa de Bases de Datos Inteligentes) que utiliza la técnica de aprendizaje numérica, es capaz de leer información de bases de datos con formato DBASE, LOTUS y ASCII. Cuenta con un manejador de bases de datos que permite crear vistas, uniones e intersecciones. En la fase de aprendizaje genera reglas que relacionan a los campos de las bases de datos, es decir, para cada campo trata de encontrar una regla matemática que lo defina. Otro software conocido es el NEUROSHELL, el cual utiliza la tecnología de redes neurales que a su vez usa la técnica de aprendizaje de propagación hacia atrás.

XI. CONCLUSIONES

- 1.- En el primer tema de interés que abordamos se discutía acerca de la diferencia entre los *lenguajes procedurales* y los *declarativos*, de lo cual pienso que no existe punto de comparación entre ellos, más bien, cada uno tiene su aplicación en áreas diferentes, los procedurales son aptos para todo tipo de sistemas incluso para el desarrollo de sistemas expertos, sin embargo, el utilizar este tipo de lenguajes para desarrollos de inteligencia artificial provoca que los tiempos de respuesta se alargen debido a la gran cantidad de búsquedas generalmente exhaustivas que se deben realizar, y no sólo esto sino que además existen bloques de instrucciones que se requieren frecuentemente, y son precisamente estos bloques los que se resumen en instrucciones más sencillas en los lenguajes declarativos, recordemos que estos fueron desarrollados sobre la base de reconocer instrucciones completas y sencillas utilizando un menor número de líneas de código.
- 2.- Otro aspecto importante que se trató es el de la definición de *programa inteligente*, aquí se recalcó que el término de *inteligencia* varía dependiendo de si se habla de *inteligencia humana* o *artificial*, esto se hace para evitar los conflictos que surgen cuando se trata de comparar ambos tipos de inteligencia, aún así, yo sigo pensando que no se puede decir que un programa es inteligente, porque sencillamente lo que hace es cumplir con las ordenes que le son dadas por medio de un conjunto de instrucciones lógicas, y la lógica de estas instrucciones está desarrollada por un programador, por un humano, por lo tanto, la inteligencia sigue siendo *inteligencia humana*.

- 3.- En cuanto al tema de *localización del aprendizaje*, creo que el capítulo fue muy claro, el aprendizaje de un sistema experto se encuentra almacenado en su base de datos, la cual puede ser desde una base de datos común hasta una red neural que por el tipo de almacenamiento que utiliza presenta ventajas sobre otros dispositivos de almacenamiento.
- 4.- En el capítulo de *clases de aprendizaje* hablamos de dos clases que yo clasifico como la base de las otras técnicas también conocidas, el *aprendizaje de memoria* y el *cognoscitivo*, pienso que de estas se derivaron las demás, el aprendizaje de memoria como ya sabemos es el más primitivo y consiste básicamente en recordar hechos específicos o secuencias de instrucciones, el cognoscitivo en cambio trata de generalizar o restringir un concepto específico, entonces como podemos ver este tipo de aprendizaje no solamente extrae información que ya tiene registrada, sino que además la analiza, la organiza y la correlaciona, y esta es precisamente la diferencia que existe entre este y el aprendizaje de memoria.
- 5.- Cuando tratamos el tema de *adquisición del conocimiento* mencionamos los tres problemas a los cuales se enfrentan las técnicas de aprendizaje que en este mismo capítulo se describieron, este tema nos dejó un catálogo de técnicas que se utilizan dependiendo de la situación del problema que tengamos enfrente, es decir, dependiendo de las necesidades del problema en sí. De esta manera, podemos elegir desde un *aprendizaje numérico* hasta un *inventivo*, sin embargo, este último siento que se encuentra aún muy lejos de alcanzarse y creo que cuando esto suceda estará todo dicho en el área de sistemas computacionales ya que los sistemas inventivos

pretenden programarse a sí mismos, y por lo tanto, seguir creciendo y desarrollándose por sí solos.

Ahora que ya conocemos las técnicas de aprendizaje existentes podemos concluir que sucedió lo que ya esperábamos, no podemos decir que existe una técnica de aprendizaje que sea la mejor, más bien, cada una tiene su aplicación para diferentes tipos de problemas dependiendo, como ya lo mencionamos, de las necesidades de estos. En resumen, no existe hasta ahora la mejor técnica de aprendizaje maquinal, es decir, no existe aún la técnica que se adapte a todo tipo de problemas y que satisfaga sus principales requerimientos. Sin embargo, no podemos decir que nunca existirá esta técnica porque aunque la posibilidad es lejana existe, y con los grandes avances que ha logrado la inteligencia artificial no dudo que algún día se logre este objetivo (el aprendizaje inventivo).

XII. APENDICE A

EJEMPLO DE REDES NEURALES

DESCRIPCION DEL EJEMPLO

El ejemplo que se muestra se desarrolló utilizando un paquete de redes neurales llamado NEUROHELL.

DEFINICION DE CARACTERISTICAS.

En esta fase se definen las características globales del problema, en el ejemplo que se presenta tenemos 6 características o nodos de entrada (orejas grandes, ojos grandes, dientes grandes, amable, arrugado y apuesto) y 7 soluciones o nodos de salida (escapar, gritar, indicar al leñador, besar la mejilla, aproximarse, ofrecer alimento y coquetear). Hasta ahora, ya definimos todos los nodos de entrada y de salida que necesitamos, la unidades o nodos ocultos son creados por el propio paquete, y estos nodos se van creando según se vaya realizando el aprendizaje, ya que hasta ahora no existe una regla que determine cual es el número más óptimo de unidades ocultas que se pueden utilizar, sólo sabemos que estas unidades facilitan un acceso más rápido de la red y que el método que actualmente se utiliza es el de la prueba y error con el cual se ha encontrado que entre más unidades ocultas se tengan el aprendizaje es más preciso, pero llegando a cierto punto este ya no avanza se queda estático, con este tipo de experimentaciones se ha encontrado además que no es posible realizar el aprendizaje completo, siempre queda cierto margen de error.

Una vez que ya definimos las características globales y las salidas del problema, necesitamos definir ahora las características específicas de cada caso y es lo que veremos a continuación.

DEFINICION DE CARACTERISTICAS

rejillas grandes
ojos grandes
dientes grandes
mable
arrugado
puesto

escapar
citar
indicar al lenador
esar la mejilla
proximarse
frezer alimento
quetear

CARACTERISTICAS DE CASOS EJEMPLO.

En esta etapa se eligen para cada caso sus características particulares de entre las globales. De esta manera estamos definiendo los tres casos que conocemos:

Caso 1.

Si el individuo tiene orejas grandes, ojos grandes y dientes grandes es un lobo, entonces Caperucita Roja debe escapar, gritar e indicar al leñador.

Caso 2.

Si es un individuo con las orejas grandes y apuesto es el leñador, entonces ella debe aproximarse, ofrecer alimento y coquetear.

Caso 3.

Si tiene los ojos grandes, es amable y está arrugado es la abuelita, entonces Caperucita debe aproximarse, besarle la mejilla y ofrecerle alimento.

Una vez que se han definido los casos estos deben ser aprendidos y es en este momento cuando se puede realizar el aprendizaje. Cuando ya se han aprendido los casos la red está preparada para responder a las preguntas que le hagamos, sin embargo, no podemos esperar que con estos 3 casos definidos las respuestas a nuestras preguntas sean 100% acertadas, y en realidad esto nunca podrá ser ni aunque definamos todos los casos habidos y por haber, por la razón que se explicaba en el punto anterior, pero, es posible disminuir el margen de error si definimos más casos ejemplo.

CARACTERISTICAS DE CASOS EJEMPLO

Example Case: 1

orejas grandes
ojos grandes
dientes grandes

escapar
gritar
indicar al lenador

Example Case: 2

orejas grandes
apuesto

aproximarse
ofrecer alimento
coquetear

Example Case: 3

ojos grandes
amable
arrugado

besar la mejilla
aproximarse
ofrecer alimento

CARACTERISTICAS DE CASOS CLASIFICADOS.

En esta fase nosotros elegimos un conjunto de características y el paquete asigna a cada salida un peso o porcentaje de certeza, de esta manera para el caso 1 les asignó el mayor peso a las salidas de escapar, gritar e indicar al leñador ya que como características de entrada nosotros seleccionamos la definición de lobo, y así para cada caso clasificado. Aquí podemos clasificar o formar todas las combinaciones que deseemos con las características globales definidas y el paquete responderá asignando pesos a las unidades de salida, donde la exactitud dependerá de la cantidad de casos definidos.

CARACTERISTICAS DE CASOS CLASIFICADOS

Classified Case: 1

orejas grandes
ojos grandes
dientes grandes

escapar 0.91
gritar 0.90
indicar al lenador 0.91

Classified Case: 2

orejas grandes
apuesto

aproximarse 0.91
ofrecer alimento 0.90
coquetear 0.90

Classified Case: 3

ojos grandes
amable
arrugado

besar la mejilla 0.90
aproximarse 0.91
ofrecer alimento 0.92

Classified Case: 4

orejas grandes
amable
arrugado

besar la mejilla 0.72
aproximarse 0.94
ofrecer alimento 0.94

Classified Case: 5

orejas grandes
ojos grandes
dientes grandes
amable
arrugado
apuesto

aproximarse 0.77
ofrecer alimento 0.75

orejas grandes
dientes grandes
apuesto

aproximarse 0.60
ofrecer alimento 0.59
coquetear 0.71

XIII. APENDICE B

EJEMPLO DE PROLOG

El ejemplo mostrado a continuación utiliza la sección goal del PROLOG y además presenta algunas de las instrucciones básicas para hacer un programa interactivo.

Básicamente, declara los siguientes predicados:

PREDICADO	SIGNIFICADO
practica(persona,deporte)	Persona practica un deporte X.
inicio	Limpia la pantalla y define una ventana.
escribe1	Envía un letrero a pantalla.
escribe2	Envía un letrero a pantalla.
escribe3(persona,deporte)	Evalúa el predicado "practica" y envía el resultado de su evaluación a pantalla.

La sección goal es la que lleva el control de la corrida del programa, como podemos observar llama a las cláusulas definidas, y estas llamadas se asemejan a las que utilizan los lenguajes procedurales para referir a una subrutina dada, sólo que en este caso las subrutinas están representadas por los predicados, los cuales pueden o no tener parámetros.

En la siguiente página se anexa el código fuente del programa además de una corrida del mismo.

* Programa ejemplo */

domains

persona, deporte = symbol

predicates

practica(persona,deporte)

inicio

escribe1

escribe2

escribe3(persona,deporte)

clauses

practica(alma,gimnasia).

practica(alma,aerobics).

practica(jose,caminata).

practica(jose,futbol).

practica(sergio,ciclismo).

practica(ricardo,natacion).

practica(ignacio,softbol).

practica(ignacio,futbol).

practica(soledad,tenis).

inicio if

clearwindow,

makewindow (1,1,112," PRUEBA ",0,0,25,80).

escribe1 if

write("Nombre de la persona: ").

escribe2 if

write("Deporte que practica: ").

escribe3(P,D) if

practica(P,D),

write(P," si practica ",D) or

not (practica(P,D)).

write(P," no practica ",D).

goal

inicio.

escribe1.

readln(Persona).

escribe2.

readln(Deporte),

escribe3(Persona,Deporte).

XIV. APENDICE C

GLOSARIO DE TERMINOLOGIA DEL AMBIENTE

El siguiente glosario incluye la descripción de conceptos utilizados en este trabajo y de los más comunes dentro del campo de la IA y que pueden ser encontrados en cualquier libro de esta área.

ABDUCCION. Es una regla de inferencia lógica, si sabemos que A implica B, y si encontramos que B es verdadero, podemos decir que A es verdadero. No obstante, esta es una forma de razonamiento heterodoxo, que permite un diagnóstico de razonamiento con "conjeturas educativas" acerca del origen de los síntomas.

ACCION. Es un elemento consecuente de una regla de producción que es ejecutado cuando la regla descarga; uno o más hechos pueden ser veraderos, por ejemplo, los procedimientos que son ejecutados, o mensajes que son impresos.

ADQUISICION DEL CONOCIMIENTO. Es un problema de aprendizaje para sistemas expertos que implica el aumento del conocimiento de un sistema que se está corriendo, donde el conocimiento es obtenido de un experto humano o creado automáticamente, con un chunking.

ALGORITMO. Es un procedimiento paso a paso que especifica exactamente como es resuelto un problema.

ALGORITMOS GENETICOS. Son técnicas de búsqueda adaptivas basadas en los principios derivados de la genética de la población natural, y están siendo aplicados actualmente a varios problemas en la ciencia, ingeniería, e IA.

ANALISIS (Parsing). Es el nivel más bajo del procesamiento del lenguaje natural, y está interesado con la decisión de sí y cómo una oración es miembro de un lenguaje examinando su estructura sintáctica.

ANALISIS DE DISERTACION. Es un nivel de procesamiento del lenguaje natural que trata de entender una sentencia y resolver las ambigüedades localizadas en el sistema de la conversación total; involucra consideraciones de intención, motivación, prevención, etc.

ANALOGIA. Método de aprendizaje y razonamiento, en el cual un problema nuevo es resuelto por igualación con una estructura similar, con problemas previamente resueltos.

APRENDIZAJE. Es la improvisación de la ejecución con experiencia sobre tiempo, y comprende mayor eficiencia, aprendizaje de memoria, inducción de categoría, adquisición del conocimiento, y descubrimiento científico.

APRENDIZAJE MAQUINAL. Es el estudio para hacer sistemas de computadoras que exhiban un comportamiento de aprendizaje.

APROXIMACION BASADA EN EL CONOCIMIENTO. Habla de la inteligencia que surge de cuerpos enormes del conocimiento en lugar de cualquier algoritmo particularmente hábil, y el modo correcto para diseñar programas inteligentes es para construir cantidades grandes de conocimiento de un dominio específico.

ARBOL. Es una gráfica acíclica dirigida con padres únicos, una estructura que consiste de nodos y arcos entre ellos acomodados en forma de un árbol con un nodo raíz único en el inicio.

ARQUITECTURA. Es la estructura básica fundamental de un sistema en procesamiento (inteligente u otro), representando el nivel de operaciones primitivas que no pueden ser cambiadas y la forma de la construcción de todas las operaciones ejecutadas por el sistema, en la mayoría de las computadoras, la organización del hardware es el nivel de la arquitectura.

BACKTRACKING (Cronológicos/Dependencia-dirigida). Es una estrategia de control que retracta las decisiones cuando avanzan a consecuencias insatisfactorias; los backtracking cronológicos van en el orden en que las elecciones fueron hechas, mientras que los de dependencia-dirigida identifican las elecciones que resultaron fallidas y van directamente hacia atrás en ellas.

BACKWARD CHAINING. Es un algoritmo de inferencia para producción de sistemas que empiecen con un hecho provisto y encuentren todas las reglas con ese hecho en sus partes de acción; entonces lo experimenta recursivamente para satisfacer las condiciones de esas reglas, los backwards trabajan desde una hipótesis hasta la prueba que resulte correcta.

BASE DEL CONOCIMIENTO. Es el almacén del conocimiento de un dominio específico de un programa de IA; en un sistema experto basado en reglas está el sistema de producción siendo usado por la máquina de inferencia.

BITMAP. Es una representación de bajo nivel de una imagen que indica la intensidad de la escala opaca de cada pixel en un arreglo de dos dimensiones representando a la imagen de entrada.

BLOCKS WORLD. Es un micromundo -con un dominio no realista del problema- usado por muchos sistemas de IA en lugar de una situación del mundo real; tiene muchas de las características

superficiales de dominios más generales pero es difícil de expandir sin complicaciones enormes.

BOTTOM-UP (análisis/procesamiento). Es una estrategia para resolución de problemas manejando datos que proceden desde los datos de entrada de un procedimiento arriba de una jerarquía -una gramática de análisis, un conocimiento con estructura de árbol basado en sistemas expertos, o la representación de un sistema visual- a la solución del problema.

BUSQUEDA BEST-FIRST. Es una técnica de búsqueda state-space que usa información heurística acerca de la deseabilidad de los estados -su distancia desde el estado objetivo- para ordenarlos por exploración en el árbol que es generado; no garantiza encontrar la solución óptima.

BUSQUEDA BLIND. Es cualquier tipo de estrategia de búsqueda no heurística y desaprensiva.

BUSQUEDA BREADTH-FIRST. Es una técnica de búsqueda que explora un nivel del árbol a un tiempo, expandiendo todos los estados en un nivel anterior moviéndose sobre el próximo; esta búsqueda garantiza terminar con una solución óptima, medida solamente por el número de estados expandidos.

BUSQUEDA HEURISTICA. Búsqueda que usa el conocimiento heurístico para enfocar la exploración del state-space en todos los posibles caminos, evitando finales inactivos, ciclos, y caminos ocultos.

CHILDREN. Son los sucesores de un nodo en un árbol, representados por figuras como nodos de más abajo de su nodo pariente común.

CHINESE ROOM. Es una metáfora usada por John Searle para describir a una computadora que está programada para mostrar un comportamiento inteligente; un no-chino puede aprender el lenguaje chino bastante bien para la gente ignorante fuera de cualquier conocimiento de lo que él ha dicho, pero él no está realmente entendiendo el lenguaje, justo como una computadora no puede realmente entender el mundo.

CHUNKING. Es un mecanismo de aprendizaje de memoria y adquisición de destreza que generalmente implica información abstracta de experiencias previas de problem-solving para crear nuevas reglas que disminuyen la cantidad de búsqueda requerida por los saltos sobre los estados inmediatos.

CIENCIA COGNOSCITIVA. Es el estudio de la inteligencia humana (uno de los objetivos de la IA) usando las técnicas y conocimientos de la psicología experimental, la ciencia de las computadoras, la neurobiología, lingüística, filosofía y antropología.

COGNICION. Es el proceso para producir comportamiento inteligente con pensamientos and/or; el principio básico de la IA y la ciencia cognoscitiva está en todos los fonemas cognoscitivos surgidos de algunas formas de computación.

COMPUTADORA DIGITAL DE PROGRAMA ALMACENADO. Es un dispositivo, esencialmente equivalente en poder computacional a la máquina de Turing, la cual obtiene sus instrucciones de la propia memoria en vez de un almacén de control separado; por lo tanto, estos programas pueden modificarse a sí mismos y ser reemplazados sin construir una nueva máquina.

CONDICION. Es un elemento en la premisa de la producción de una regla; regularmente, las condiciones son conjuntas, tanto que todas deben ser satisfechas por la regla causada.

CONEXIONISMO. Es un sistema y metodología para modelos cognoscitivos que representan el procesamiento de información en un nivel fundamentalmente más bajo que la manipulación tradicional de símbolos, un nivel cuyas unidades primitivas se asemejan a las neuronas y conexiones de los símbolos y el nivel más alto de la estructura no forma parte de ellos.

CONJUNTO CONFLICTO. Es una estructura que colecta todas las reglas en un sistema de producción que puede descargar en cualquier ciclo del sistema; esto es, todas las producciones cuyas condiciones antecedentes son satisfechas por el contenido actual de memoria trabajando.

CONOCIMIENTO. Es cualquier información útil para que una tarea sea ejecutada.

CONOCIMIENTO DECLARATIVO. Es el conocimiento ordinario acerca de los conceptos y las relaciones entre ellos, independiente de cualquier procedimiento que los manipule.

CONOCIMIENTO DE SENTIDO COMUN. Es el conocimiento ordinario acerca del mundo y la proporción que sostiene a otros, el conocimiento del dominio específico; esta es la clase de conocimiento declarativo que necesitamos usualmente para representar las redes semánticas y sistemas frame.

CONOCIMIENTO PROCEDURAL. Es el conocimiento acerca de como ejecutar varias preguntas, usualmente representadas por varios procedimientos que las hacen.

DEDUCCION. Es un proceso seguro de inferencia lógica que usa la siguiente regla: dado que A implica B, si sabemos que A es verdadero, podemos concluir que B también lo es; esta es una regla

segura, llamada *modus ponens*, la cual sustenta virtualmente a todos los argumentos lógicos y experimentados.

DEPENDENCIA CONCEPTUAL (CD). Es una teoría de desarrollo de semántica para representar todo el conocimiento en aplicaciones de entendimiento del lenguaje natural que está basada en un sistema de once hechos en donde todos pueden ser descompuestos.

DESCARGA. De una regla que ha sido disparada para ser seleccionada por una aplicación y tiene una parte de la acción ejecutada.

DOMINIO. Es una área o campo de experiencia y conocimiento de un humano o programa de computadora, que delimita la clase de problemas a los que el sistema puede ser aplicado.

ELEMENTO DEL APRENDIZAJE. Es la porción de aprendizaje de un sistema de IA que decide como modificar el elemento de ejecución e implementar esas modificaciones; normalmente incluye cualquier algoritmo de aprendizaje empleado por el sistema.

ELIMINACION DE CANDIDATOS. Es un procedimiento de aprendizaje para descripción de conceptos singulares que remueve desde el conjunto de posibles conceptos a todos los candidatos que no cubren un ejemplo positivo nuevo, o a los que cubren un ejemplo negativo nuevo.

EPAM (Elementary Perceiver And Memorizer). Sistema desarrollado por Feigenbaum y Simon para aprendizaje de conceptos simples y categorías basado en una red de discriminación que se produce con ejemplos positivos nuevos.

ESTRATEGIA DE RESOLUCION DE CONFLICTOS. Es un método predefinido para decidir cual regla en un sistema de producción se descarga en

un ciclo , cuando más de una son creadas (el conjunto conflicto contiene más de una regla); la regla que se selecciona es la que tiene el factor de certeza más alto, el más alto en un orden preestablecido, o el primero en entrar al conjunto conflicto.

EXPLOSION COMBINATORIA. Es el problema fundamental de cualquier sistema basado en la búsqueda que limita su utilidad práctica con problemas grandes, en los cuales el número de alternativas a explorar se incrementa tan rápido como los progresos de la búsqueda; por ejemplo, una búsqueda en un árbol con una ramificación de factor 5 puede solamente ser explorado a un nivel profundo por una computadora rápida a 5 tiempos.

FACTOR DE CERTEZA (CF). Es un valor numérico, frecuentemente en el rango de -1.0 a +1.0, con el 0.0 indicando no conocimiento, que representa el grado de certeza fijo a un hecho o conclusión; las reglas de inferencia acertadas pueden calcular la certeza de inferencias basadas en la certeza de sus premisas.

FILLER. Es un valor que ocupa un slot en un frame, frecuentemente representado como un apuntador hacia otro frame en un sistema frame.

FILTRACION. Es un proceso usado por los sistemas de percepción para modificar la entrada, en contenido pero no en formato, en el orden en que remueve ruido y destaca los lineamientos y regularidades importantes; visualmente, esto consiste en la aplicación de varias funciones matemáticas para cada punto en un bitmap.

FRAME. Es una estructura de datos, similar a un registro en una base de datos relacional, la cual representa un concepto con un nombre y varias propiedades preparadas en pares slot/filler.

GENERACION DE CANDIDATOS. Es un procedimiento usado en los diagnósticos de sistemas de razonamiento que reduce el conjunto de componentes con posibles fallas por generación de sospechas y entonces cualquiera demuestra o no demuestra que estos están fallando.

GRAMATICA. Es cualquier sistema formal que describe la estructura sintáctica de un lenguaje, usualmente por establecimiento de varias categorías de palabras y secuencias, además las reglas que restringen a esas secuencias pueden ser combinadas con oraciones del lenguaje.

GRAMATICA DE CONTEXTO LIBRE (CFG). Es una gramática constituida por símbolos terminales, símbolos no terminales, y reglas de reescritura que tienen un no-terminal singular en su lado izquierdo y un string de cualquier símbolo en su lado derecho; por consiguiente, la reescritura legal del símbolo no-terminal no se hace dependiente de los símbolos próximos.

GRAMATICA DE CONTEXTO SENSITIVO (CSG). Es una gramática idéntica a la gramática de contexto libre pero cuyas reglas de reescritura pueden contener ambos símbolos terminal y no-terminal en su lado izquierdo.

GRAMATICA REGULAR. Es una gramática de contexto libre restringida, que conduce sólo a reglas con un símbolo no-terminal singular sobre el lado derecho, y es bastante poderosa para representar lenguajes regulares.

HEURISTICA. Es una regla de desaprobación, una pieza de información que puede proveer una guía útil para resolver un problema pero que no garantiza obtener resultados buenos en todas las situaciones en contraste con un algoritmo, el cual por

diseños especificados tiene un procedimiento que siempre trabajará.

IA FIRME. Es el nombre dado por John Searle a un programa de investigación cuyo objetivo es la creación de un pensamiento artificial completo que actualmente los estados cognoscitivos entenderían y experimentarían de algún modo al pensamiento humano.

INDUCCION. Es el proceso de aprendizaje y formulación de nuevos conceptos dando solamente ejemplos (positivos y/o negativos) de ellos.

INFERENCIA BAYESIANA. Es un medio por el cual un sistema de razonamiento puede usar el Teorema de Bayes para manipular información incierta usando las probabilidades condicional y la anterior, una de las dos solamente o en combinación con un sistema basado en reglas; esta es una técnica más formal que los factores de certeza usados en MYCIN y sus derivados.

INGENIERIA DEL CONOCIMIENTO. Es la práctica de construcción de un sistema experto por extracción del dominio y el conocimiento de un problema específico de un experto y la representación en una forma adecuada para una implementación en una computadora.

INHERITANCIA. Habilita un concepto, tal como "perro", en una red semántica o un sistema frame para adquirir automáticamente las propiedades de la clase superior, tal como "animal" y "mamífero", sin tener explícitamente anexado el concepto mismo; trabaja por exploración ascendiendo en las instancias y en los encadenamientos de las subclases para encontrar el concepto antecesor en la jerarquía.

INTELIGENCIA ARTIFICIAL. Es una ciencia que estudia principalmente las técnicas para hacer que las computadoras exhiban un comportamiento inteligente, del mismo modo que los procesos cognoscitivos humanos y la construcción de un sistema artificial autónomo, generalmente inteligente.

INTERFASE DEL LENGUAJE NATURAL (NLI). Es un programa usado como un frente final en un programa de aplicación, tal como un manejador de bases de datos o un sistema experto que permite al usuario comunicarse con la aplicación en un subconjunto de un lenguaje natural como el Inglés en lugar del lenguaje especial y los comandos del software.

LISP (Procesando lista). Desarrollado por John McCarthy alrededor de 1960, es el lenguaje dominante de programación para la investigación y desarrollo de IA por las facilidades para manipular los símbolos y las listas en estructuras sofisticadas.

MAQUINA DE TURING. Es un modelo formal de computación que está máximamente generalizado en el que cualquier proceso computacional posible puede ser ejecutado por una máquina de Turing construida apropiadamente.

MEMORIA TRABAJANDO. En un sistema de producción que tiene datos transientes acerca del problema que actualmente está siendo resuelto, tal como hechos dados o deducidos; en psicología esto es llamado memoria de término-corto, considerando que la memoria de término-grande almacena el sistema actual de las reglas de producción y los procedimientos que ellas usan.

MODELO COGNOSCITIVO. Es una aproximación para crear comportamiento inteligente por procesos de simulación conocidos o confiables que ocurren en el cerebro humano, como en contra de la creación de

procedimientos que se parecen a "hacer el acontecimiento justo" desatendiendo su autenticidad mental.

OPERACIONALIZACION. Es el proceso de implementación en el progreso del descubrimiento del elemento de ejecución por el elemento aprendido en un sistema de IA; puede en algún momento ser cercado por el truco de la representación singular, el cual garantiza que el conocimiento está en un formato consistente e intercambiable.

PERCEPCION. Es el proceso de recibir una simulación continua desde el medio ambiente, en formas como luz para vision o sonido para el entendimiento del lenguaje, y derivando sus contenidos semánticos relevantes construyendo una representación conceptual de estas.

PRAGMATICA. Es el área de la lingüística que estudia las interacciones entre las oraciones para entender ambigüedades como la referencia de pronombres y sentidos determinados; también, un nivel de análisis de diálogo en el procesamiento del lenguaje natural que busca resolver los problema desarrollados por tales ambigüedades.

PREMISA. Es el antecedente If-part sobre el lado izquierdo de una regla de producción que lista las condiciones bajo las cuales la regla es aplicable y en la cual la satisfacción es necesaria antes que la regla pueda ser usada.

PROBLEMA DE ASIGNACION DE CREDITO. En el aprendizaje es el que determina cual procedimiento en el elemento ejecutado contribuyó a la solución correcta y cual no, tal que el elemento aprendido pueda modificar la porción apropiada del sistema.

PROBLEMA FRAME. Es una representación notoria y un problema de razonamiento que cuestiona como un sistema puede saber cuales elementos del conocimiento llegan inválidos en los pasos del tiempo, como cuando un robot se mueve en un lugar nuevo, y debe de algún modo olvidar el lugar previo.

PROCEDIMIENTO BRITISH-MUSEUM. Es un algoritmo de búsqueda horriblemente ineficiente que genera movimientos u operadores aleatorios hasta que llega a la solución deseada del problema.

PROCEDIMIENTO DE DECISION. Es un algoritmo preciso que toma como entrada un string de símbolos y decide si es o no miembro de un lenguaje; por lo tanto, un procedimiento particular de decisión es empleado para definir un lenguaje no finito.

PROGRAMACION AUTOMATICA. Es un subcampo de la IA y un software de ingeniería que estudia a los sistemas que puedan por sí mismos escribir nuevos programas de computadoras o modificar los ya existentes.

PROGRAMACION LOGICA (LP). Es el uso de la lógica como un lenguaje de programación, en donde los programas consisten de axiomas y control ejercidos por un algoritmo comprobado por teoremas.

PROLOG (Programming in Logic). Es el lenguaje de programación lógica más popular usado para aplicaciones de IA, especialmente en procesos de lenguaje natural con una gramática fácil de cláusulas definidas y en los sistemas expertos con su demostrador del teorema de la cadena de retroceso.

RECURSION. Ocurre cuando un procedimiento se ejecuta por sí mismo, llamándose directamente o llamando a otro procedimiento que a su vez llama al original, después de que ha hecho sus cálculos; es

contrastado con la iteración, cuando un procedimiento se cicla a sí mismo.

RED DE DISCRIMINACION. Es un dispositivo con estructura de árbol que clasifica los objetos en categorías de ramas hasta llegar a la fronda que nombre a la clase propia.

RED DE TRANSICION AUMENTADA (ATND). Es un dispositivo de procesamiento y un formalismo para representar la estructura sintáctica de un lenguaje complejo, especialmente lenguaje natural; esto es una red de transición recursiva con la adición de registros y pruebas.

RED DE TRANSICION RECURSIVA. Es un dispositivo que analiza oraciones con gramática de contexto libre representando el conjunto de reglas para reescribir cada símbolo no-terminal como un autómata finito separado que puede hacer saltos recursivos a otro autómata como la condición para saltos a un nuevo estado.

RED NEURAL. En biología es una malla interconectada de células nerviosas; en IA denota una red de procesamiento paralelo distribuido modelada estrechamente en las propiedades de las neuronas reales.

RED SEMANTICA. Es un formalismo de la representación del conocimiento que presenta cada concepto como un nodo en una gráfica y las relaciones entre los conceptos como arcos entre los nodos; el sentido de un concepto es derivado de su relación con los otros conceptos en la red.

REGLA DE PRODUCCION. Es un If-then, asociación premisa-acción usada para representar ambos conocimientos declarativo y procedural necesarios para resolver los problemas.

REGLA DE REESCRITURA. Es una regla en una gramática que permite el reemplazamiento de una combinación de símbolos por otra tal como generar oraciones del lenguaje (top-down) o reducir una oración dada al símbolo raíz S (bottom-up).

REPRESENTACION DEL CONOCIMIENTO. Es la ciencia de codificación del conocimiento del mundo real en un formato eficiente que lo hace fácil para los programas que lo usan.

ROBOTICA. Es un programa de investigación que estudia la construcción de agentes artificiales autónomos, completamente inteligentes y hábiles para percibir su medio ambiente y sus movimientos justo como un ser humano.

SEMANTICA. Es el nivel de análisis en el procesamiento de un lenguaje natural que trata del sentido de las oraciones, el cual se deriva del sentido de las palabras y frases componentes y del contexto circunvecino.

SIMBOLO NO TERMINAL. En una gramática no aparece en las oraciones del lenguaje generadas por esta, pero representa una categoría o combinación legal de símbolos en las reglas de reescritura de la gramática, tal como NP para "noun phrase" en Inglés.

SIMBOLO TERMINAL. Es un caracter del alfabeto de un lenguaje particular el cual aparece como un string que está en las oraciones del lenguaje, tal como "a" o "b" en Inglés.

SINTAXIS. Es el nivel de análisis en el procesamiento de un lenguaje que decide si una oración está en un lenguaje y como están hechos los lineamientos estructurales de un miembro del lenguaje; interactúa fuertemente con los niveles de semántica y pragmática.

SISTEMA BASADO EN EL CONOCIMIENTO (KBS). Es un sistema que tiene un comportamiento inteligente usando algoritmos bastante simples con una estructura del conocimiento específica y de sentido común, en lugar de aplicar algoritmos poderosos pero complejos con información de un dominio específico pequeño.

SISTEMA DE PRODUCCION. Es una colección organizada de reglas de producción que trabajan juntas para resolver uno o más problemas bajo un algoritmo de inferencia apropiado.

SISTEMA EXPERTO (SE). Es un sistema basado en el conocimiento capaz de ejecutar algún problema restringido con un dominio comparable a o mejor que el de un experto humano; frecuentemente tiene la habilidad para explicar las conclusiones e interactivamente adquirir conocimiento nuevo.

SISTEMA EXPERTO SHELL. Es un SE "vacío" con todos los componentes pero con el dominio específico del conocimiento base; el cual puede ser creado por un experto humano sin experiencia de IA; usualmente provee guía interactiva para simplificar el proceso de adquisición del conocimiento inicial.

SISTEMA FRAME. Es una jerarquía hereditaria de frames, incluyendo los valores default y el procedimiento enlazado, para hacer un sistema completo que represente el conocimiento declarativo.

SLOT. Es un elemento en un frame que representa una propiedad o lineamiento del objeto, cuyo valor es contenido en el filler para el slot.

STATE SPACE. Es una representación del dominio de un problema como un conjunto de estados, o configuraciones de los elementos del problema, y operadores, o manipulaciones que convierten un estado

en otro, que puede ser dibujado como una gráfica dirigida en la cual los nodos son estados y los arcos son operadores.

TOP-DOWN (Análisis/Procesamiento). Es la aproximación a la característica de un sistema analítico que empieza con una descripción de nivel alto de la entrada que será reconocida, tal como una expectación de lo que sería, e intenta acceder los datos de la hipótesis (si es posible); si falla, puede tratar con diferentes hipótesis, posiblemente derivadas de procesos bottom-up preliminares.

UNIDADES. En el conexionismo, redes con procesamiento paralelo distribuido, son altamente interconectadas, procesadores muy pequeños que calculan funciones simples de sus señales de entrada desde otras unidades para producir un resultado de salida similar.

VISION. Es el proceso de recibir, transducir, y entender imágenes inicialmente representadas como luz y finalmente como una estructura conceptual apropiada y útil.

XV. REFERENCIAS

TITULO: NeuroShell

HARDWARE COMPATIBLE: IBM PC y compatibles

SISTEMA OPERATIVO REQUERIDO: PC-DOS/MS-DOS

MEMORIA REQUERIDA: 256k

NOTA: Sistema experto que imita el aprendizaje biológico humano.

AUTOR: Thinking Software, Inc.; Think Soft

TITULO: Building blocks for speech Modular neural networks are a new approach to high-performance speech recognition.

AUTOR: Waibel, Alex; Hampshire

FUENTE: BYTE , Agosto 1/1989.

TITULO: Neural Computers: NATO ASI Series F, Vol. 41

AUTOR: Eckmiller, Rolf, ed

FUENTE: Computer Book Review , May 1988 , v6

TITULO: Neural-network heuristics: Three heuristic algorithms that learn from experience

AUTOR: Josin, Gary

FUENTE: BYTE , Oct 1987 , v12

TITULO: Back-propagation: A generalized delta learning rule

AUTOR: Jones, William P; Hoskins, Josiah

FUENTE: BYTE , Oct 1987 , v12

TITULO: An artificial neural network experiment: This program acts as a learning machine

AUTOR: Brown, Robert Jay

FUENTE: Dr. Dobb's Journal of Software Tools , Apr 1987 , v12

TITULO: CyberTools Artificial Intelligence Library

HARDWARE COMPATIBLE: IBM PC y compatibles, PC AT

SISTEMA OPERATIVO REQUERIDO: PC-DOS, MS-DOS

LENGUAJES: Pascal, BASIC, C, Assembly, LISP, Prolog

MEMORIA REQUERIDA: 640k

NOTA: Colección de programas para estudiantes de IA y sistemas expertos. Redes neurales, la percepción de la red, propagación hacia atrás, etc.

AUTOR: CyberToolWorks; CyberTool

TITULO: Analysis of potential for new places of business
(decision support system)

AUTOR: Van Beek, J. J. J.

AUTOR AFILIADO: BSO/Manage. Support BV, Utrecht, Netherlands
Sol, H.G.; Takkenberg, C.A.Th.; De Vries Robbe,
P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in
Decision Support Systems. Proceedings of the Second
Mini Euroconference. 1987.

TITULO: Knowledge acquisition from user programs by computer
with own knowledge-based system

AUTOR: Kazimierzak, J.

AUTOR AFILIADO: Inst. of Eng. Cybern., Tech. Univ. Wroclaw,
Poland Sol, H.G.; Takkenberg, C.A.Th.; De
Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987

TITULO: A knowledge based approach to business planning

AUTOR: Gallo, P.; Massa-Rolandino, R.; Canepa, F.; Borazzo, F.

AUTOR AFILIADO: Artificial Intelligence Lab., CSI-Piemonte,
Torino, Italy Sol, H.G.; Takkenberg, C.A.Th.;
De Vries Robbe, P.F. (Editors) Sponsor: Shell
Nederland; IBM Nederland; Hewlett Packard
Nederland; et al

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: DECIDEX, a multi-expert system for strategic decisions

AUTOR: Levine, P.; Maillard, J.C.; Pomerol, J.C.

AUTOR AFILIADO: Dept. Inf., ENA, Paris, France Sol, H.G.;
Takkenberg, C.A.Th.; De Vries Robbe, P.F.
(Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: ASPES: A skeletal Pascal expert system

AUTOR: Doukidis, G.I.; Paul, R.J.

AUTOR AFILIADO: Dept. of Stat. & Math. Sci., London Sch. of
Econ. & Political Sci., England Sol, H.G.;
Takkenberg, C.A.Th.; De Vries Robbe, P.F.
(Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Extensions to the expert system shell DELFI-2

AUTOR: Lucas, P.J.F.; de Swaan Arons, H.

AUTOR AFILIADO: Centre for Math. & Comput. Sci., Amsterdam,
Netherlands Sol, H.G.; Takkenberg, C.A.Th.; De
Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Microcomputer based expert system shells-the
spreadsheets of artificial intelligence

AUTOR: O'Keefe, R.; Belton, V.; Ball, T.

AUTOR AFILIADO: Rutherford Coll., Kent Univ., Canterbury,
England Sol, H.G.; Takkenberg, C.A.Th.; De
Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Heuristics in the Abbot-Monk problem

AUTOR: van den Herik, H.J.; Stoop, J.C.; Varkevisser, R.P.

AUTOR AFILIADO: Dept. of Math. & Inf., Delft Univ. of
Technol., Netherlands Sol, H.G.; Takkenberg,
C.A.Th.; De Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Modelling support environments

AUTOR: Maschtera, U.

AUTOR AFILIADO: Dept. of Comput. Sci., J.K. Univ., Linz,
Austria Sol, H.G.; Takkenberg, C.A.Th.; De
Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Linguistic exploratory analysis for decision support

AUTOR: Koppelaar, H.

AUTOR AFILIADO: Dept. of Math. & Inf., Tech. Univ., Delft,
Netherlands Sol, H.G.; Takkenberg, C.A.Th.; De
Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Knowledge-based formulation of linear planning models

AUTOR: Binbasioglu, M.; Jarke, M.

AUTOR AFILIADO: Graduate Sch. of Bus. Admin., New York Univ.,
New York, NY, USA Sol, H.G.; Takkenberg,
C.A.Th.; De Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second - Mini
Euroconference. 1987.

TITULO: A logic programming approach to building planning and
simulation models

AUTOR: Lee, R.M.

AUTOR AFILIADO: Graduate Sch. of Bus., Texas Univ., Austin, TX,
USA Sol, H.G.; Takkenberg, C.A.Th.; De Vries
Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Responsive decision support systems: a broad view
illustrates when to include expert systems

AUTOR: Kurstedt, H.A., Jr.

AUTOR AFILIADO: Dept. of Ind. Eng. & Oper. Res., Virginia
Polytech. Inst. & State Univ., Blacksburg, VA,
USA Sol, H.G.; Takkenberg, C.A.Th.; De Vries
Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in
Decision Support Systems. Proceedings of the Second
Mini Euroconference. 1987.

TITULO: Decision support systems and expert systems: a complementary
relationship

AUTOR: Pfeifer, R.; Lithi, H.-J.

AUTOR AFILIADO: Dept. of Comput. Sci., Zurich Univ.,
Switzerland Sol, H.G.; Takkenberg, C.A.Th.;
De Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Decision support systems: a discipline or a vision

AUTOR: Bosman, A.

AUTOR AFILIADO: Inf. Syst. Res. Group, Groningen Univ.,
Netherlands Sol. H.G.; Takkenberg, C.A.Th.;
De Vries Robbe, P.F. (Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Models: operational, expert and intelligent

AUTOR: Hertz, D.B.

AUTOR AFILIADO: Intelligent Comput. Syst. Res. Inst., Miami
Univ., CoralGables, FL, USA Sol. H.G.;
Takkenberg, C.A.Th.; De Vries Robbe, P.F.
(Editors)

FUENTE: Expert Systems and Artificial Intelligence in Decision
Support Systems. Proceedings of the Second Mini
Euroconference. 1987.

TITULO: Interpolation results in the split inversion neural network algorithm

AUTOR: McAulay, A.D.; Ravula, R.

AUTOR AFILIADO: Dept. of Comput. Sci. & Eng., Wright State Univ., Dayton, OH, USA

FUENTE: Proceedings of the IEEE 1989 National Aerospace and Electronics Conference NAECON 1989

TITULO: Neural network design considerations for EEG spike detection

AUTOR: Eberhart, R.C.; Dobbins, R.W.; Webber, W.R.S.

AUTOR AFILIADO: Appl. Phys. Lab., Johns Hopkins Univ., Laurel, MD, USA Buus, S. (Editors)

FUENTE: Proceedings of the Fifteenth Annual Northeast Bioengineering Conference. 1989.

TITULO: A P-ternary threshold element network—an application of ternary logic to a neural system treating ambiguity

AUTOR: Yamamoto, Y.; Mukaidono, M.

AUTOR AFILIADO: Fac. of Econ., Takasaki City Univ. of Econ., Japan

FUENTE: Proceedings. The Nineteenth International Symposium on Multiple-Valued Logic. 1989.

TITULO: Neural network control of unknown nonlinear systems

AUTOR: Li, W.; Slotine, J.-J.E.

AUTOR AFILIADO: Nonlinear Syst. Lab., MIT, Cambridge, MA, USA

FUENTE: Proceedings of the 1989 American Control Conference. 1989.

TITULO: Neural networks for system identification

AUTOR: Chu, R.; Shoureshi, R.; Tenorio, M.

AUTOR AFILIADO: Purdue Univ., West Lafayette, IN, USA

FUENTE: Proceedings of the 1989 American Control Conference.
1989.

TITULO: Adaptive control of unknown dynamical systems via
neural network approach

AUTOR: Lan, M.-S.

AUTOR AFILIADO: Rockwell Int. Sci. Center, Thousand Oaks, CA,
USA

FUENTE: Proceedings of the 1989 American Control Conference.
1989.

TITULO: Decision-making with the Boltzmann machine

AUTOR: Kam, M.; Cheng, R.

AUTOR AFILIADO: Dept. of Electr. & Comput Eng., Drexel Univ.,
Philadelphia, PA, USA

FUENTE: Proceedings of the 1989 American Control Conference.
1989.

TITULO: A learning strategy for the control of a one-legged
hopping robot

AUTOR: Helferty, J.J.; Collins, J.B.; Wong, L.C.; Kam, M.

AUTOR AFILIADO: Dept. of Electr. Eng., Temple Univ.,
Philadelphia, PA, USA

FUENTE: Proceedings of the 1989 American Control Conference

TITULO: Experimental learning in an optical perceptron like neural network

AUTOR: Yoshinaga, H.; Kitayama, K.; Hori, T.

AUTOR AFILIADO: NTT Transmission Syst. Labs., Yokosuka, Japan

FUENTE: Opt. Lett. (USA) vol.14, no.14. 1989.

TITULO: A neural net model that learns spectrum patterns of five vowels and its discrimination mechanisms

AUTOR: Ohtomo, T.; Hara, K.

AUTOR AFILIADO: Fac. of Eng., Yamagata Univ., Yonezawa, Japan

FUENTE: Trans. Inst. Electron. Inf. Commun. Eng. D-II (Japan) vol.J72D-II, no.6. Junio 1989.

TITULO: Pattern recognition of the ^1H NMR spectra of sugar alditols using a neural network

AUTOR: Thomsen, J.U.; Meyer, B.

AUTOR AFILIADO: Dept. of Biochem., Georgia Univ., Athens, GA, USA

FUENTE: J. Magn. Reson. (USA) vol.84, no.1, Agosto 1989.

TITULO: Learning in feedforward layered networks: the tiling algorithm

AUTOR: Mezard, M.; Nadal, Jean-P.

AUTOR AFILIADO: Lab. de Phys. Theorique, Ecole Normale Superieure, Paris, France

FUENTE: J. Phys. A, Math. Gen. (UK) vol.22, no.12. Junio 21 1989.

TITULO: Phase transitions in simple learning

AUTOR: Hertz, J.A.; Krogh, A.; Thorbergsson, G.I.

AUTOR AFILIADO: Nordita, Copenhagen, Denmark

FUENTE: J. Phys. A, Math. Gen. (UK) vol.22, no.12. Junio 21 1989.

TITULO: Perceptron learning with sign-constrained weights

AUTOR: Amit, D.J.; Wong, K.Y.M.; Campbell, C.

AUTOR AFILIADO: Dept. of Phys., Imperial Coll., London, UK

FUENTE: J. Phys. A, Math. Gen. (UK) vol.22, no.12. Junio 21
1989.

TITULO: Training with noise and the storage of correlated
patterns in a neuralnetwork model

AUTOR: Gardner, E.J.; Wallace, D.J.; Stroud, N.

AUTOR AFILIADO: Dept. of Phys., Edinburgh Univ., UK

FUENTE: J. Phys. A, Math. Gen. (UK) vol.22, no.12. Junio 21
1989.

TITULO: Retrieval properties of a neural network with an
asymmetric learning rule

AUTOR: Gardner, E.; Mertens, S.; Zippelius, A.

AUTOR AFILIADO: Dept. of Phys., Edinburgh Univ., UK

FUENTE: J. Phys. A, Math. Gen. (UK) vol.22, no.12. Junio 1989.

TITULO: Neurocomputer and robot control

AUTOR: Asakawa, K.

FUENTE: J. Soc. Instrum. Control Eng. (Japan) vol.28, no.3.
Marzo 1989.

TITULO: An adaptive least squares algorithm for the
efficient training of artificial neural networks

AUTOR: Kollias, S.; Anastassiou, D.

AUTOR AFILIADO: Dept. of Electr. Eng., Columbia Univ., NY, USA

FUENTE: IEEE Trans. Circuits Syst. (USA) vol.36, no.8. Agosto
1989.

TITULO: A high-storage capacity content-addressable memory
and its learning algorithm

AUTOR: Verleysen, M.; Sirletti, B.; Vandemeulebroecke, A.;
Jespers, P.G.A.

AUTOR AFILIADO: Lab. de Microelectron., Univ. Catholique
de Louvain, Louvain-la-Neuve, Belgium

FUENTE: IEEE Trans. Circuits Syst. (USA) vol.36, no.5. Mayo
1989.

TITULO: Learning of fast transforms and spectral domain neural
computing

AUTOR: Ersoy, O.K.; Chen, C.-H.

AUTOR AFILIADO: Sch. of Electr. Eng., Purdue Univ., West
Lafayette, IN, USA

FUENTE: IEEE Trans. Circuits Syst. (USA) vol.36, no.5. Mayo
1989.

TITULO: Learning and optimization of machining operations
using computingabilities of neural networks

AUTOR: Rangwala, S.S.; Dornfeld, D.A.

AUTOR AFILIADO: Dept. of Mech. Eng., California Univ.,
Berkeley, CA, USA

FUENTE: IEEE Trans. Syst. Man Cybern. (USA) vol.19, no.2.
Marzo-Abril 1989.

TITULO: Neural learning of constants nonlinear transformations

AUTOR: Barhen, J.; Gulati, S.; Zak, M.

AUTOR AFILIADO: Jet Propulsion Lab., California Inst. of
Technol., Pasadena, CA, USA

FUENTE: Computer (USA) vol.22, no.6. Junio 1989.

TITULO: A comparison of processor topologies for a fast trainable neural network for speech recognition

AUTOR: Suzuki, Y.; Atlas, L.E.

AUTOR AFILIADO: NTT Human Interface Lab., Kanagawa, Japan

FUENTE: IEEE ICASSP-89: 1989 International Conference on Acoustics, Speech and Signal Processing. 1989.

TITULO: Deconvolution and nonlinear inverse filtering using a neural network

AUTOR: Glanz, F.H.; Miller, W.T. *

AUTOR AFILIADO: Dept. of Electr. & Comput. Eng.,
New Hampshire Univ., Durham, NH, USA

FUENTE: ICASSP-89: 1989 International Conference on Acoustics, Speech and Signal Processing vol.4. 1989.

TITULO: An analysis of a noise reduction neural network

AUTOR: Tamura, S.

AUTOR AFILIADO: ATR Interpreting Telephony Res. Lab., Kyoto,
Japan

FUENTE: ICASSP-89: 1989 International Conference on Acoustics, Speech and Signal Processing vol.3. 1989.

TITULO: Neural network learning paradigms involving nonlinear spectral processing

AUTOR: Ersoy, O.K.; Hong, D.

AUTOR AFILIADO: Sch. of Electr. Eng., Purdue Univ., West
Lafayette, IN, USA

FUENTE: ICASSP-89: 1989 International Conference on Acoustics, Speech and Signal Processing vol.3. 1989.

TITULO: Object recognition using a neural network and invariant Zernike features

AUTOR: Khotanzad, A.; Lu, J.H.

AUTOR AFILIADO: Dept. of Electr. Eng., Southern Methodist Univ., Dallas, TX, USA

FUENTE: Proceedings CVPR '89 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1989.

TITULO: Analog electronic neural network circuits

AUTOR: Graf, H.P.; Jackel, L.D.

AUTOR AFILIADO: AT&T Bell Lab., Holmdel, NJ, USA

FUENTE: IEEE Circuits Devices Mag. (USA) vol.5, no.4. Julio 1989.

TITULO: Neural networks for high-storage content-addressable memory: VLSI circuit and learning algorithm

AUTOR: Verleysen, M.; Sirletti, B.; Vandemeulebroecke, A.M.; Jespers, P.G.A.

AUTOR AFILIADO: Lab. of Microelectron., Univ. Catholique de Louvain, Louvain-la-Neuve, Belgium

FUENTE: IEEE J. Solid-State Circuits (USA) vol.24, no.3. Junio 1989.

TITULO: Image compression using a multilayer neural network

AUTOR: Luttrell, S.P.

AUTOR AFILIADO: R. Signals & Radar Establ., Malvern, UK

FUENTE: Pattern Recognit. Lett. (Netherlands) vol.10, no1. Julio 1989.

TITULO: A capacitive neural network for associative memory

AUTOR: Verleysen, M.; Martin, D.; Jespers, P.

AUTOR AFILIADO: Lab. of Microelectron., Univ. Catholique
de Louvain, Louvain-la-Neuve, Belgium Barbe,
A.M. (Editors)

FUENTE: Proceedings of the Tenth Symposium on Information
Theory in the Benelux. 1989.

TITULO: A learning mechanism for invariant pattern recognition
in neural networks

AUTOR: Coolen, A.C.C.; Kuijk, F.W.

AUTOR AFILIADO: Dept. of Med. & Physiol. Phys., Utrecht Univ.,
Netherlands Barbe, A.M. (Editors)

FUENTE: Proceedings of the Tenth Symposium on Information
Theory in the Benelux. 1989.

TITULO: A Marquardt learning algorithm for neural networks

AUTOR: De Wilde, P.

AUTOR AFILIADO: Dept. of Math., Katholieke Univ. Leuven,
Belgium Barbe, A.M. (Editors)

FUENTE: Proceedings of the Tenth Symposium on Information
Theory in the Benelux. 1989.

TITULO: Learning phoneme recognition using neural networks

AUTOR: Renals, S.; Rohwer, R.

AUTOR AFILIADO: Edinburgh Univ., UK

FUENTE: ICASSP-89: 1989 International Conference on Acoustics,
Speech and Signal Processing vol.1. 1989.

TITULO: Shift-invariant, multi-category phoneme recognition
using Kohonen's LVQ2

AUTOR: McDermott, E.; Katagiri, S.

AUTOR AFILIADO: ATR Auditory & Visual Perception Res. Lab.,
Osaka, Japan

FUENTE: ICASSP-89: 1989 International Conference on Acoustics,
Speech and Signal Processing vol.1. 1989.

TITULO: Speaker-independent word recognition using dynamic
programming neural networks

AUTOR: Sakoe, H.; Isotani, R.; Yoshida, K.; Iso, K.-I.;
Watanabe, T.

AUTOR AFILIADO: NEC Corp., Kawasaki, Japan

FUENTE: ICASSP-89: 1989 International Conference on Acoustics,
Speech and Signal Processing vol.1. 1989.

TITULO: A simple neuron feature detection

AUTOR: Hambaba, M.L.

AUTOR AFILIADO: Dept. of Electr. Eng. & Comput. Sci.,
Stevens Inst. of Technol., Hoboken, NJ, USA

FUENTE: Eighth Annual International Phoenix Conference on
Computers and Communications. 1989 Conference
Proceedings. 1989.

TITULO: A neural network learning strategy for the
control of a one-legged hopping machine

AUTOR: Helferty, J.J.; Collins, J.B.; Kam, M.

AUTOR AFILIADO: Dept. of Electr. Eng., Temple Univ.,
Philadelphia, PA, USA

FUENTE: Proceedings. 1989 IEEE International Conference on
Robotics and Automation vol.3. 1989.

TITULO: Behavior control for a mobile robot by
multihierarchical neural network

AUTOR: Sekiguchi, M.; Negata, S.; Asakawa, K.

AUTOR AFILIADO: Fujitsu Lab. Ltd., Kawasaki, Japan

FUENTE: Proceedings. 1989 IEEE International Conference on
Robotics and Automation vol.3. 1989.

TITULO: Learning with nonstatic paradigms in neural networks

AUTOR: Kohle, M.; Schonbauer, F.

AUTOR AFILIADO: Tech. Univ. of Vienna, Austria

FUENTE: MIV-89 Proceedings of the International Workshop on
Industrial Applications of Machine Intelligence and
Vision (Seiken Symposium). 1989.

TITULO: Syntactic category disambiguation with neural networks

AUTOR: Benello, J.; Mackie, A.W.; Anderson, J.A.

AUTOR AFILIADO: Dept. of Cognitive & Linguistic Sci.,
Brown Univ., Providence, RI, USA

FUENTE: Comput. Speech Lang. (UK) vol.3, no.3. Julio 1989.

TITULO: Classification performance of a Hopfield neural
network based on aHebbian-like learning rule

AUTOR: Jacyna, G.M.; Malaret, E.R.

AUTOR AFILIADO: UNISYS Corp., Reston, VA, USA

FUENTE: IEEE Trans. Inf. Theory (USA) vol.35, no.2. Marzo
1989.

TITULO: A parallel network that learns to play backgammon

AUTOR: Tesauro, G.; Sejnowski, T.J.

AUTOR AFILIADO: Center for Complex Syst. Res., Illinois
Univ., Urbana Champaign, IL, USA

FUENTE: Artif. Intell. (Netherlands) vol.39, no.3. Julio 1989.

TITULO: Applications of neural networks to consumer products

AUTOR: Pawson, R.

FUENTE: IEE Colloquium on 'Current Issues in Neural Network Research'(Digest No.83). 1989.

TITULO: Applying adaptive training methods to networks with Boolean functions

AUTOR: Martland, D.

AUTOR AFILIADO: Dept. of Comput. Sci., Brunel Univ., Uxbridge, UK

FUENTE: IEE Colloquium on 'Current Issues in Neural Network Research'(Digest No.83). 1989.

TITULO: An artificial neural network for classification of forced expired volumesignals

AUTOR: Gage, H.D.; Miller, T.K.

AUTOR AFILIADO: Dept. of Electr. & Comput. Eng., North Carolina State Univ., Raleigh, NC, USA Harris, G.; Walker, C. (Editors)

FUENTE: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society vol.3. 1988.

TITULO: Nestor Decision Learning System for mortgage underwriting and delinquencyrisk assessment

AUTOR: Buffa, M.

AUTOR AFILIADO: Nestor Inc., Providence, RI, USA

FUENTE: Second European Seminar on Neural Computing: Commercial Prospects. 1989.

TITULO: Text processing for speech synthesis using parallel distributed models

AUTOR: Scordilis, M.S.; Gowdy, J.N.

AUTOR AFILIADO: Dept. of Electr. & Comput. Eng., Clemson Univ., SC, USA

FUENTE: SOUTHEASTCON '89 Proceedings. Energy and Information Technologies in the Southeast vol.2. 1989.

TITULO: Fault tolerance of neural networks

AUTOR: Damarla, T.R.; Bhagat, P.K.

AUTOR AFILIADO: Kentucky Univ., Lexington, KY, USA

FUENTE: SOUTHEASTCON '89 Proceedings. Energy and Information Technologies in the Southeast vol.1. 1989.

TITULO: Implementation of a programmable artificial neuron using discrete logic

AUTOR: Kwon, T.M.; Valdez, M.E.

AUTOR AFILIADO: Dept. of Comput. Eng., Minnesota Univ., Duluth, MN, USA

FUENTE: SOUTHEASTCON '89 Proceedings. Energy and Information Technologies in the Southeast vol.1. 1989.

TITULO: A hierarchical sigma-pi neural architecture for learning search algorithms

AUTOR: Gee-gwo Mei; Wentai Liu; Su-shing Chen

AUTOR AFILIADO: Dept. of Electr. & Comput. Eng., North Carolina State Univ., Raleigh, NC, USA Ras, Z.W.; Saitta, L. (Editors)

FUENTE: Methodologies for Intelligent Systems, 3. Proceedings of the Third International Symposium. 1988.

TITULO: Using a context-sensitive learning for robot arm control

AUTOR: Yeung, D.-Y.; Gekey, G.A.

AUTOR AFILIADO: Dept. of Comput. Sci., Univ. of Southern California, Los Angeles, CA, USA

FUENTE: Proceedings. 1989 IEEE International Conference on Robotics and Automation vol.3. 1989.

TITULO: Pruning versus clipping in neural networks

AUTOR: Janowsky, S.A.

AUTOR AFILIADO: Dept. of Phys., Harvard Univ., Cambridge, MA, USA

FUENTE: Phys. Rev. A, Gen. Phys. (USA) vol.39, no.12. Junio 1989.

TITULO: Neural network model for fast learning and retrieval

AUTOR: Arsenault, H.H.; Macukow, B.

AUTOR AFILIADO: Dept. de Phys., Laval Univ., Ste-Foy, Que., Canada

FUENTE: Opt. Eng., Bellingham (USA) vol.28, no.5. Mayo 1989.

TITULO: Introduction to neural computation

AUTOR: Cammarata, S.

FUENTE: Sist. Impresa (Italy) vol.35, no.302. Abril 1989.

TITULO: Associative memory in neural networks with the Hebbian learning rule

AUTOR: Tsodyks, M.V.

AUTOR AFILIADO: Inst. of Higher Nervous Activity & Neurophysiol., Acad. of Sci., Moscow, USSR

FUENTE: Mod. Phys. Lett. B (Singapore) vol.3, no.7. Mayo 1989.

TITULO: Neural network principles for theoretical psychology

AUTOR: Levine, D.S.

AUTOR AFILIADO: Texas Univ., Arlington, TX, USA

FUENTE: Behav. Res. Methods Instrum. Comput. (USA) vol.21,
no.2. Abril 1989.

TITULO: Profiles in learning (neural networks)

AUTOR: Passarelli, B.

FUENTE: Unix Rev. (USA) vol.7, no.5. Mayo 1989.

TITULO: Neuron networks-motor control

AUTOR: Matsuo, K.

AUTOR AFILIADO: Int. Inst. for Adv. Study of Social Inf. Sci.,
Fujitsu Ltd., Tokyo, Japan

FUENTE: J. Jpn. Soc. Simul. Technol. (Japan) vol.8, no.1.
Marzo 1988.

TITULO: Neuron networks-pattern recognition

AUTOR: Yamada, K.; Kami, H.

AUTOR AFILIADO: NEC Corp., Tokyo, Japan

FUENTE: J. Jpn. Soc. Simul. Technol. (Japan) vol.8, no.1.
Marzo 1989.

TITULO: Neuron networks-associative memory

AUTOR: Hirai, Y.

AUTOR AFILIADO: Inst. of Inf. Sci. & Electron., Tsukuba Univ.,
Ibaraki, Japan

FUENTE: J. Jpn. Soc. Simul. Technol. (Japan) vol.8, no.1.
Marzo 1989.

TITULO: Introduction-neurocomputing

AUTOR: Matsumoto, G.

FUENTE: J. Jpn. Soc. Simul. Technol. (Japan) vol.8, no.1.
Marzo 1989.

TITULO: Optimising PSS using learning automata

AUTOR: Mars, P.

AUTOR AFILIADO: Sch. of Eng. & Appl. Sci., Durham Univ., UK

FUENTE: IEE Colloquium on 'Advances in Optimisation' (Digest No.45). 1989.

TITULO: Simulation of a neural network solution to the media selection problem

AUTOR: Clarson, V.H.; Shimp, J.E.

AUTOR AFILIADO: E-Syst. Inc., St. Petersburg, FL, USA Webster, W. (Editors)

FUENTE: Simulation and AI, 1989. Proceedings of the SCS Western Multiconference. 1989.

TITULO: Neural network models and applications to speech processing. II

AUTOR: Bourlard, H.; Wellekens, C.J.

AUTOR AFILIADO: Philips Res. Lab., Brussels, Belgium

FUENTE: Rev. HF (Belgium) vol.14, no.3-4. 1988.

TITULO: Adaptive moving object tracking integrating neural networks and intelligent processing

AUTOR: Lee, J.S.J.; Nguyen, D.D.; Lin, C.

AUTOR AFILIADO: Boeing Electron. High Technol. Center, Seattle, WA, USA

FUENTE: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.1002. 1989.

TITULO: Self-organising neural network for binary image recognition

AUTOR: Allinson, N.M.; Johnson, M.J.

AUTOR AFILIADO: Dept. of Electron., York Univ., UK

FUENTE: IEE Colloquium on 'Pattern Recognition for Binary Images' (Digest No.53). 1989.

TITULO: Decomposition of neural network models of robot dynamics: a feasibility study

AUTOR: Bassi, D.F.; Bekey, G.A.

AUTOR AFILIADO: Dept. of Comput. Sci., Univ. of Southern California, Los Angeles, CA, USA Webster, W. (Editors)

FUENTE: Simulation and AI, 1989. Proceedings of the SCS Western Multiconference. 1989.

TITULO: Digital neural networks

AUTOR: Martinez, T.R.

AUTOR AFILIADO: Dept. of Comput. Sci., Brigham Young Univ., Provo, UT, USA

FUENTE: Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics vol.1. 1988.

TITULO: Comparison of continuous and discrete learning through a geometric representation

AUTOR: Pemberton, J.C.

FUENTE: Issued by: Univ. California, Comput. Sci. Dept., Los Angeles, CA, USA. Enero-Marzo 1989.

TITULO: Feedback-error-learning neural network for trajectory control of a robotic manipulator

AUTOR: Miyamoto, H.; Kawato, M.; Setoyama, T.; Suzuki, R.

AUTOR AFILIADO: Fac. of Eng. Sci., Osaka Univ., Japan

FUENTE: Neural Netw. (USA) vol.1, no.3. 1988.

TITULO: Analysis of hidden units in a layered network trained to classify sonar targets

AUTOR: Gorman, R.P.; Sejnowski, T.J.

FUENTE: Neural Netw. (USA) vol.1, no.1. 1988.

TITULO: Teaching feed-forward neural networks by simulated annealing

AUTOR: Engel, J.

AUTOR AFILIADO: Normal Bridge Lab. of Phys., California Inst. of Technol., Pasadena, CA, USA

FUENTE: Complex Syst. (USA) vol.2, no.6. Diciembre 1988.

TITULO: What can neural networks do?

AUTOR: Uesaka, Y.

AUTOR AFILIADO: Fac. of Sci. & Technol., Sci. Univ. of Tokyo, Japan

FUENTE: J. Inst. Electron. Inf. Commun. Eng. (Japan) vol.71, no.11. Noviembre 1988.

TITULO: A programmable analog neural network chip

AUTOR: Schwartz, D.B.; Howard, R.E.; Hubbard, W.E.

AUTOR AFILIADO: AT&T Bell Lab., Holmdel, NJ, USA

FUENTE: IEEE J. Solid-State Circuits (USA) vol.24, no.2. Abril 1989.

TITULO: Schemas and neural networks for sixth generation computing

AUTOR: Arbib, M.A.

AUTOR AFILIADO: Center for Neural Eng., Univ. of Southern California, Los Angeles, CA, USA

FUENTE: J. Parallel Distrib. Comput. (USA) vol.6, no.2. Abril 1989.

TITULO: Real-time neuromorphic algorithms for inverse kinematics of redundant manipulators

AUTOR: Barhen, J.; Gulati, S.; Zak, M.

AUTOR AFILIADO: Jet Propulsion Lab., California Inst. of Technol., Pasadena, CA, USA

FUENTE: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.1002. 1989.

TITULO: Practical demonstration of a learning control system for a five-axis industrial robot

AUTOR: Hewes, R.P.; Miller, W.T., III

AUTOR AFILIADO: Dept. of Electr. & Comput. Eng., New Hampshire Univ., NH, USA

FUENTE: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.1002. 1989.

TITULO: Neural network applications in hand-written symbol understanding

AUTOR: Kangsuk Lee

AUTOR AFILIADO: Siemens Corp. Res., Princeton, NJ, USA

FUENTE: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.1002. 1989.

TITULO: Pattern recognition using a CMAC based learning system

AUTOR: Herold, D.J.; Miller, W.T.; Kraft, L.G.; Glanz, F.H.

AUTOR AFILIADO: New Hampshire Univ., Durham, NH, USA

FUENTE: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.1004. 1989.

TITULO: Principal components analysis of images via back propagation

AUTOR: Cottrell, G.W.; Munro, P.

AUTOR AFILIADO: California Univ., San Diego, La Jolla, CA, USA

FUENTE: Source: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.1001. 1988.

TITULO: Neural-space generalization of a topological transformation

AUTOR: Josin, G.

AUTOR AFILIADO: Neural Syst. Inc., Vancouver, BC, Canada

FUENTE: Biol. Cybern. (West Germany) vol.59, no.4-5. 1988.

TITULO: Accelerating the convergence of the back-propagation method

AUTOR: Vogl, T.P.; Mangis, J.K.; Rigler, A.K.; Zink, W.T.; Alkon, D.L.

AUTOR AFILIADO: Environ. Res. Inst. of Michigan, Arlington, VA, USA

FUENTE: Biol. Cybern. (West Germany) vol.59, no.4-5. 1988.

TITULO: Neural network architecture for robot hand control

AUTOR: Liu, H.; Iberall, T.; Bekey, G.A.

AUTOR AFILIADO: Dept. of Comput. Sci., Univ. of Southern California, Los Angeles, CA, USA

FUENTE: IEEE Control Syst. Mag. (USA) vol.9, no.3. Abril 1989.

TITULO: An architecture for very large neural networks with high connectivity

AUTOR: Allinson, N.M.; Johnson, M.J.

AUTOR AFILIADO: Dept. of Electron., York Univ., UK

FUENTE: IEE Colloquium on 'VLSI and Architectures for Symbolic Processing' (Digest No.37). 1989.

TITULO: An adaptable parallel search of knowledge bases with beam search

AUTOR: Presern, S.; Brajak, P.; Vogel, L.; Zeleznikar, A.P.

AUTOR AFILIADO: Iskra Delta Comput., Ljubljana, Yugoslavia
Blanning, R.; King, D. (Editors)

FUENTE: Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences.
Vol.III: Decision Support and Knowledge Based Systems Track (Cat. No.89TH0244-4) vol.3. 1989.

TITULO: A network of multistate units capable of associative memory and patternclassification

AUTOR: Banzhaf, W.

AUTOR AFILIADO: Inst. fur Theor. Phys. und Synergetik,
Stuttgart Univ., West Germany

FUENTE: Physica D (Netherlands) vol.34, no.3. Marzo 1989.

TITULO: A hierarchical neural network model for category detection

AUTOR: Sakaguchi, H.

AUTOR AFILIADO: Dept. of Phys., Kyushu Univ., Fukuoka, Japan

FUENTE: Prog. Theor. Phys. (Japan) vol.81, no.2. Febrero 1989.

TITULO: Applications of counterpropagation networks

AUTOR: Hecht-Nielsen, R.

AUTOR AFILIADO: Hecht-Nielsen Neurocomput. Corp., San Diego,
CA, USA

FUENTE: Neural Netw. (USA) vol.1, no.2. 1988.

TITULO: Neocognitron: a hierarchical neural network capable of visual patternrecognition

AUTOR: Fukushima, K.

AUTOR AFILIADO: NHK Sci. & Tech. Res. Labs., Tokyo, Japan

FUENTE: Neural Netw. (USA) vol.1, no.2. 1988.

TITULO: Neural network applications to speech processing

AUTOR: Shikano, K.; Nakamura, M.; Tamura, S.; Waibel, A.

AUTOR AFILIADO: ATR Interpreting Telephony Res. Labs., Osaka,
Japan

FUENTE: J. Acoust. Soc. Jpn. (Japan) vol.44, no.10. Octubre 1988.

TITULO: Beyond artificial intelligence. The wonders of neural network computers

AUTOR: Anderson, S.M.

FUENTE: Unisphere (USA) vol.9, no.1. Abril 1989.

TITULO: Learning by diffusion for multilayer perceptron

AUTOR: Hopcroft, R.G.; Hall, T.J.

AUTOR AFILIADO: Dept. of Phys., King's Coll. London., UK

FUENTE: Electron. Lett. (UK) vol.25, no.8. Abril 1989.

TITULO: Connectionist approach for robot grasp planning

AUTOR: Ali, A.L.; Ali, K.S.; Ali, D.L.

AUTOR AFILIADO: Dept. of Comput. Sci. & Comput. Eng. Technol., Univ. of Southern Mississippi, Hattiesburg, MS, USA

FUENTE: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.1008. 1989.

TITULO: Adaptive learning with hidden units using a single photorefractive crystal

AUTOR: Peterson, C.; Redfield, S.

AUTOR AFILIADO: Microelectron. & Comput. Technol. Corp., Austin, TX, USA

FUENTE: Proc. SPIE - Int. Soc. Opt. Eng. (USA) vol.963. 1989.

TITULO: Application of an intelligent signal processing system to acoustic emission analysis

AUTOR: Grabec, I.; Sachse, W.

AUTOR AFILIADO: Dept. of Theor. & Appl. Mech., Cornell Univ., Ithaca, NY, USA

FUENTE: J. Acoust. Soc. Am. (USA) vol.85, no.3. Marzo 1989.

TITULO: An optical learning machine

AUTOR: Farhat, N.H.; Shae, Z.-Y.

AUTOR AFILIADO: Dept. of Electr. Eng., Pennsylvania Univ.,
Philadelphia, PA, USA Hoevel, L.W.;
Milutinovic, V. (Editors)

FUENTE: Proceedings of the Twenty-Second Annual Hawaii
International Conference on System Sciences. Vol.I:
Architecture Track vol.1. 1989.

TITULO: Controlling basins of attraction in a neutral
network-based telemetry monitor

AUTOR: Bell, B.; Eilbert, J.L.

AUTOR AFILIADO: Electromagnetics Inst., Tech. Univ. of
Denmark, Lyngby, Denmark

FUENTE: Fourth Conference on Artificial Intelligence
for Space Applications (NASA Conf. Publ. 3013). 1988.

TITULO: Dynamics of learning in simple perceptrons

AUTOR: Hertz, J.A.; Thorbergsson, G.I.; Krogh, A.

AUTOR AFILIADO: Nordita, Copenhagen, Denmark

FUENTE: Phys. Scr. Vol. T (Sweden) vol.T25. 1989.

TITULO: Pattern retrieval and learning in nets of
asynchronous binary threshold elements

AUTOR: Kam, M.; Cheng, R.; Guez, A.

AUTOR AFILIADO: Dept. of Electr. & Comput. Eng., Drexel
Univ., Philadelphia, PA, USA

FUENTE: IEEE Trans. Circuits Syst. (USA) vol.36, no.3. Marzo
1989.

TITULO: Optimal size of neural networks with multiplicative learning

AUTOR: Nemeth, R.; Geszti, T.

AUTOR AFILIADO: Dept. of Atom. Phys., Roland Eotvos Univ.,
Budapest, Hungary

FUENTE: Acta Phys. Hung. (Hungary) vol.64, no.1-3. 1989.

TITULO: Coordinates transformation and learning control
for visually-guided voluntary movement with iteration:
a Newton-like method in a function space

AUTOR: Kawato, M.; Isobe, M.; Maeda, Y.; Suzuki, R.

AUTOR AFILIADO: Dept. of Biophys. Eng., Osaka Univ., Toyonaka,
Japan

FUENTE: Biol. Cybern. (West Germany) vol.59, no.3. 1988.

TITULO: Fractal Chaos: a new neural network holographic model

AUTOR: Perez, J.-C.

AUTOR AFILIADO: IBM France Montpellier, France

FUENTE: Proceedings SEAS Anniversary Meeting 1988 vol.2.
1988.

TITULO: The foundations of automatic learning and
generalization

AUTOR: Denker, J.

AUTOR AFILIADO: AT&T Bell Lab., Holmdel, NJ, USA Myklebust,
J.B.; Harris, G.F. (Editors)

FUENTE: Proceedings of a Special Symposium on Maturing
Technologies and Emerging Horizons in Biomedical
Engineering. 1988.

TITULO: Using the ADAP learning algorithm to forecast the onset of diabetesmellitus

AUTOR: Smith, J.W.; Everhart, J.E.; Dickson, W.C.; Knowler, W.C.; Johannes, R.S.

AUTOR AFILIADO: Nat. Inst. of Diabetes, Digestive & Kidney Diseases, Bethesda, MD, USA Greenes, R.A. (Editors)

FUENTE: Proceedings. The Twelfth Annual Symposium on Computer Applications in Medical Care. 1988.

TITULO: Determination of testing efficacy in carcinoma of the lung using a neuralnetwork model

AUTOR: Hudson, D.L.; Cohen, M.E.; Anderson, M.F.

AUTOR AFILIADO: California Univ., San Francisco, CA, USA Greenes, R.A. (Editors)

FUENTE: Proceedings. The Twelfth Annual Symposium on Computer Applications in Medical Care. 1988.

TITULO: The estimation of posterior probabilities using neural networks (medicalcomputing)

AUTOR: Hripcsak, G.

AUTOR AFILIADO: Columbia Presbyterian Med. Center, New York, NY, USA Greenes, R.A. (Editors)

FUENTE: Proceedings. The Twelfth Annual Symposium on Computer Applications in Medical Care. 1988.

TITULO: A neural network architecture for cue-based motion planning

AUTOR: Zacksenhouse, M.; deFigueiredo, R.J.P.; Johnson, D.H.

AUTOR AFILIADO: Dept. of Electr. & Comput. Eng., Rice Univ., Houston, TX, USA

FUENTE: Proceedings of the 27th IEEE Conference on Decision and Control vol.1. 1988.

TITULO: WSI fault-tolerant semantic network architecture

AUTOR: Delgado-Frias, J.G.; Moore, W.R.

EDITADA POR: Univ. Oxford, UK. Diciembre 1988.

TITULO: A short-term neural network memory

AUTOR: Morris, R.J.T.; Wing Shing Wong

AUTOR AFILIADO: AT&T Bell Labs., Holmdel, NJ, USA

FUENTE: SIAM J. Comput. (USA) vol.17, no.6. Diciembre 1988.

TITULO: The least constraint principle for learning in
neurodynamics

AUTOR: Zak, M.

AUTOR AFILIADO: Jet Propulsion Lab., California Inst. of
Technol., Pasadena, CA, USA

FUENTE: Phys. Lett. A (Netherlands) vol.135, no.1. Febrero
1989.

TITULO: Neural networks-application-specific problem solvers

AUTOR: Wilson, A.C.; Wilson, D.

FUENTE: ESD, Electron. Syst. Des. Mag. (USA) vol.19, no.2.
Febrero 1989.

TITULO: Neural-network technology and its applications

AUTOR: Roth, M.W.

AUTOR AFILIADO: Appl. Phys. Lab., Johns Hopkins Univ., Laurel,
MD, USA

FUENTE: Johns Hopkins APL Tech. Dig. (USA) vol.9, no.3.
Julio-Septiembre 1988.

TITULO: A unified framework for connectionist systems

AUTOR: Golden, R.M.

AUTOR AFILIADO: Dept. of Pshychol., Standford Univ., CA, USA

FUENTE: Biol. Cybern. (West Germany) vol.59, no.2. 1988.

TITULO: Spatial organization and genetic information in brain development

AUTOR: Gierer, A.

AUTOR AFILIADO: Max-Planck-Inst. fur Entwicklungsbiol.,
Tubingen, West Germany

FUENTE: Biol. Cybern. (West Germany) vol.59, no.1. 1988.

TITULO: Self-organising multilayer topographic mappings

AUTOR: Luttrell, S.P.

AUTOR AFILIADO: R. Signals & Radar Establ., Malvern, UK

FUENTE: IEEE International Conference on Neural Networks
vol.1. 1988.

TITULO: IEEE International Conference on Neural
Networks (IEEE Cat.No.88CH2632-8)

Sponsor: IEEE

Publication Year: 1988

Conference Information :

IEEE International Conference on Neural Networks (IEEE
No.88CH2632-8) Date: 24-27 July 1988 Location: San Diego, CA, USA

Publ: IEEE, New York, NY, USA. 2 vol. (699+651) pp.

TITULO: Neurocomputer-controlled stepper motor balances a
broom

AUTOR: Sasseen, R.

FUENTE: Powerconvers. Intell. Motion (USA) vol.14,no.,1.
Diciembre 1988.

TITULO: Neural network learning of spectral features of
nonverbal speech

AUTOR: Lerner, S.Z.; Deller, J.R.

AUTOR AFILIADO: Dept. of Electr. Comput. Eng., Northeastern
Univ., Boston, MA, USA LaCourse, J.R. (Editors)

FUENTE: Proceedings of the Fourteenth Annual Northeast
Bioengineering Conference. 1988.

TITULO: Mandarin syllable recognition system with learning ability based on neural network model

AUTOR: Jhing-Fa Wang; Chung-Hsien Wu; Jau-Yien Lee; Chia-Nien Wang

AUTOR AFILADO: Inst. of Electr. & Comput. Eng., Nat. Cheng Kung Univ., Tainan, Taiwan

FUENTE: 1988 International Conference on Computer Processing of Chinese and Oriental Languages. Proceedings. 1988.

TITULO: A neural network approach to Chinese information retrieval

AUTOR: Chan, S.C.; Low, H.B.; Yu, W.C.P.

AUTOR AFILIADO: Nat. Univ. of Singapore, Kent Ridge, Singapore

FUENTE: 1988 International Conference on Computer Processing of Chinese and Oriental Languages. Proceedings. 1988.

TITULO: Connectionist recruitment learning

AUTOR: Diederich, J.

AUTOR AFILIADO: Int. Comput. Sci. Inst., Berkeley, CA, USA
Kodratoff, Y. (Editors)

FUENTE: ECAI 88. Proceedings of the 8th European Conference on Artificial Intelligence. 1988.

TITULO: Robot control using neural networks

AUTOR: Josin, G.; Charney, D.; White, D.

AUTOR AFILIADO: Neural Syst. Inc., Vancouver, BC, Canada

FUENTE: IEEE International Conference on Neural Networks. vol.2. 1988.

TITULO: Neural networks mediating linearizable dynamic
redundant sensori-motor reflexes characterized by
minimum of Hermitian norm

AUTOR: Daunicht, W. J.

AUTOR AFILIADO: Dept. of Biophys., Dusseldorf Univ., West
Germany

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: A neuromorphic controller with a human teacher

AUTOR: Guez, A.; Selinsky, J.

AUTOR AFILIADO: Dept. of Electron. Comput. Eng., Drexel
Univ., Philadelphia, PA, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: A learning architecture for control based on
back-propagation neural networks

AUTOR: Elsley, R. K.

AUTOR AFILIADO: Rockwell Int. Sci. Center, Thousand Oaks, CA,
USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Building a generic architecture for robot hand control

AUTOR: Liu Huan; Iberall, T.; Bekey, G. A.

AUTOR AFILIADO: Dept. of Comput. Sci., Univ. of Southern
California, Los Angeles, CA, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Inverse Viterbi algorithm as learning procedure
and application to optimization in the string
instrument fingering problem

AUTOR: Sayegh, S.I.; Tenorio, M.F.

AUTOR AFILIADO: Dept. of Phys., Indiana-Purdue Univ., Fort
Wayne, IN, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Neural networks in defense applications

AUTOR: Castelaz, P.F.

AUTOR AFILIADO: Hughes Aircraft Co., Fullerton, CA, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: An application of a multiple neural network learning
system to emulation of mortgage underwriting judgements

AUTOR: Collins, E.; Ghosh, S.; Scofield, C.

AUTOR AFILIADO: Nestor Inc., Providence, RI, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Economic prediction using neural networks: the case
of IBM daily stock returns

AUTOR: White, H.

AUTOR AFILIADO: Dept. of Econ., California Univ., San Diego,
CA, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Color analysis by learning subspaces and optical processing

AUTOR: Parkkinen, J.; Oja, E.; Jaaskelainen, T.

AUTOR AFILIADO: Dept. of Comput. Sci. & Phys., Kuopio Univ.,
Finland

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Bimodal stochastic optical learning machine

AUTOR: Farhat, N.H.; Shae, Z.Y.

AUTOR AFILIADO: Dept. of Electr. Eng., Pennsylvania Univ.,
Philadelphia, PA, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Adaptive networks as a model for human speech development

AUTOR: Tenorio, M.F.; Tom, M.D.; Schwartz, R.G.

AUTOR AFILIADO: Sch. of Electr. Eng., Purdue Univ., West
Lafayette, IN, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Creation by refinement: a creativity paradigm for gradient descent learning networks

AUTOR: Lewis, J.P.

AUTOR AFILIADO: Digital Sound Lab., New York Inst. of
Technol., Old Westbury, NY, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Learning to understand sentences in a connectionist network

AUTOR: Nolfi, S.; Parisi, D.

AUTOR AFILIADO: Fondazione Sigma Tau, Roma, Italy

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Psychologically plausible features for shape recognition in a neuralnetwork

AUTOR: Krishnan, G.; Walters, D.

AUTOR AFILIADO: Dept. of Comput. Sci., State Univ. of New York, Buffalo, NY, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: Neural network models and their application to handwritten digitrecognition

AUTOR: Pawlicki, T.F.; Dan-Shyang Lee; Hull, J.J.; Srihari, S.N.

AUTOR AFILIADO: Dept. of Comput. Sci., State Univ. of New York, Buffalo, NY, USA

FUENTE: IEEE International Conference on Neural Networks
vol.2. 1988.

TITULO: The resonance correlation network

AUTOR: Ryan, T.W.

AUTOR AFILIADO: Sci. Applications Int. Corp., Tucson, AZ, USA

FUENTE: IEEE International Conference on Neural Networks
vol.1. 1988.

