

Redes Neuronales Artificiales

Jose Aguilar
Cemisid, Facultad de Ingeniería
Universidad de los Andes
Mérida, Venezuela
aguilar@ula.ve

contenido

- Introducción
- Aprendizaje
- Modelos de redes neuronales artificiales

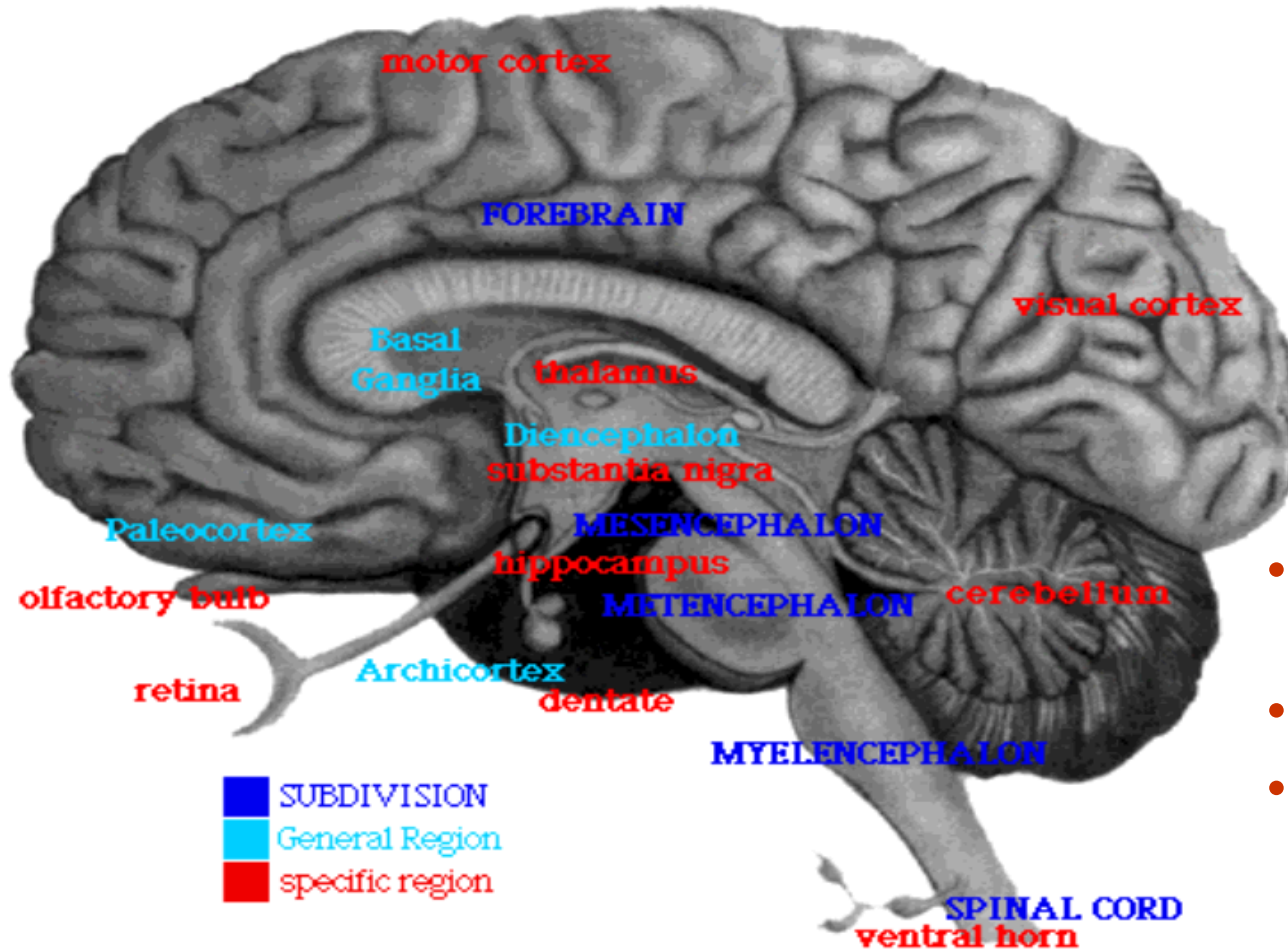
REDES NEURONALES ARTIFICIALES

Las Redes Neuronales Artificiales están basadas en el funcionamiento de las neuronas biológicas que componen el cerebro de los animales. El cerebro es el órgano central del sistema nervioso animal y contiene de 50 a 100 mil millones de neuronas con funciones especializadas, de las cuales aproximadamente 10 mil millones son células piramidales corticales, las cuales transmiten señales por medio de 1000 billones de conexiones sinápticas. Recibe información proveniente de los sentidos, la cual es procesada con una gran velocidad mediante la combinación de la información almacenada dando respuestas que controlan y regulan las acciones y reacciones del cuerpo.

INTRODUCCIÓN A LAS RNAs

***EL SISTEMA NEURONAL TIENE COMO PROPOSITO EL
CONTROL CENTRALIZADO DE VARIAS FUNCIONES
BIOLOGICAS, POR EJEMPLO EL ABASTECIMIENTO
DE ENERGIA, FUNCIONES MOTORAS Y
SENSORIALES, ETC.***

Cerebro Humano



- 10^{11} Neuronas (procesadores)
- Poder desconocido
- 1000 – 10000 conecciones por neurona

MODELO BIOLOGICO

- ***SISTEMA NEURONAL***

***=> CONTROL CENTRALIZADO DE
FUNCIONES BIOLOGICAS***

- ***CEREBRO = 10 BILLONES DE NEURONAS Y
60 TRILLONES DE CONEXIONES***

MODELO BIOLOGICO

- ***NEURONAS: CELULAS VIVAS***
- ***CARACTERISTICAS:***
 - ***ELEMENTOS SIMPLES INTERCONECTADOS***
 - ***FUNCIONAMIENTO EN PARALELO, ASINCRÓNICAS Y NO ALGORÍTMICAS***
 - ***INTERACCIONES COMPLEJAS***

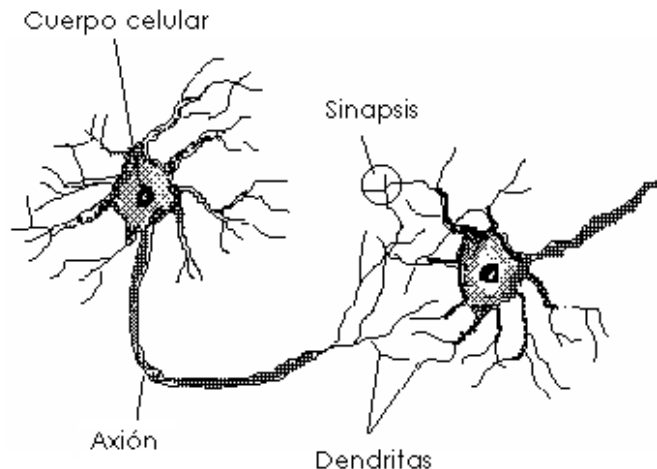
. NEURONA BIOLÓGICA

Una neurona biológica está conformada por tres elementos principales:

Soma o cuerpo de la neurona el cual contiene el núcleo y tiene forma piramidal o esférica, mide entre 10 y 80 micras y es en este lugar donde se ejecutan las transformaciones necesarias para la vida de la neurona.

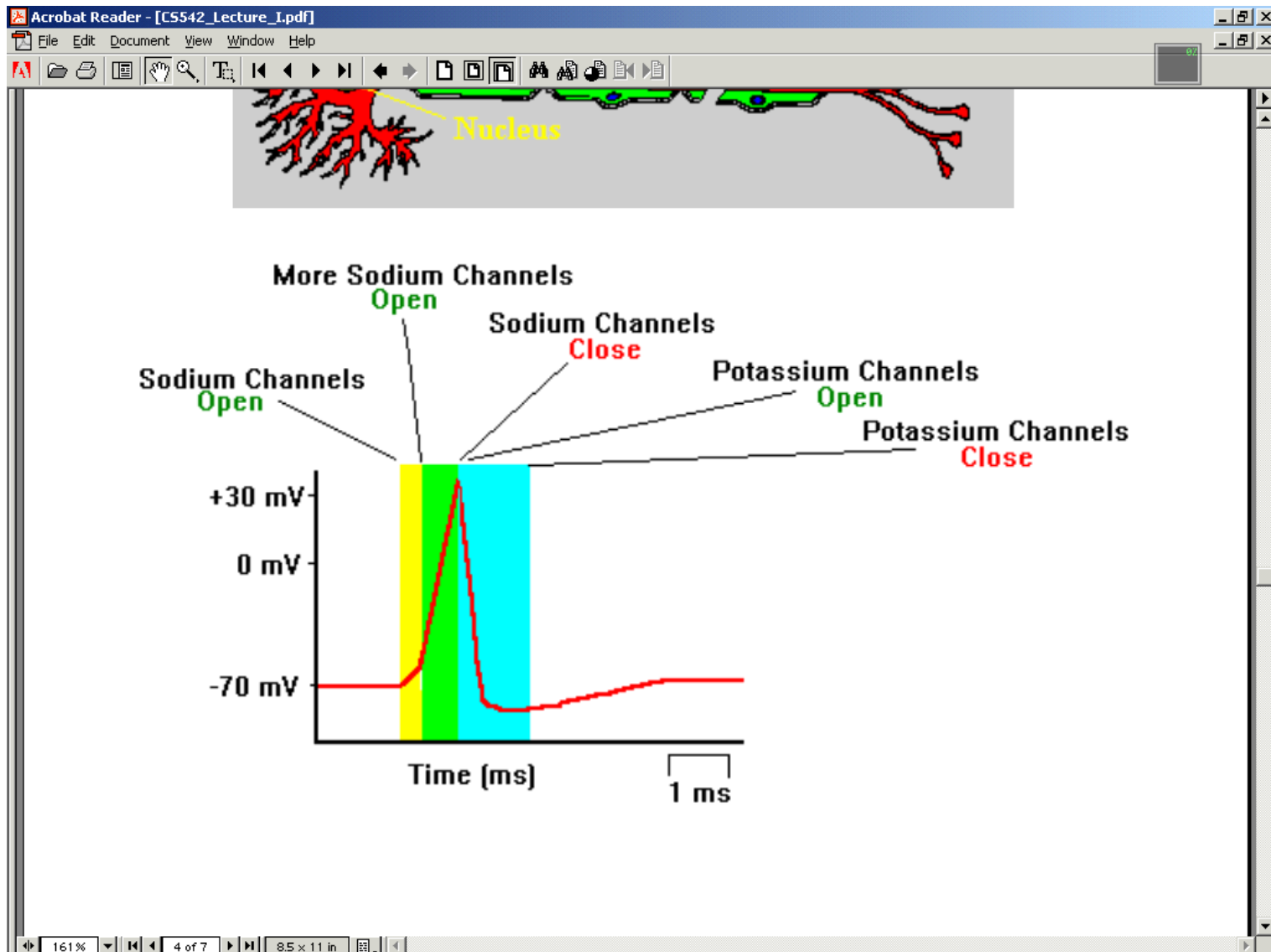
Axón es el elemento que lleva la salida de información de la neurona a las dendritas de otras neuronas. Es una prolongación larga y delgada, mide desde 10 micras hasta 1 m de longitud, y conduce la información desde el cuerpo celular hacia las dendritas de otras neuronas.

Dendritas o ramas de extensión utilizadas para recibir las entradas de información, son un conjunto de ramificaciones que salen del cuerpo celular de la neurona. Son receptoras y reciben señales eléctricas a través de conexiones sinápticas.



La sinapsis es una estructura especial donde se forma la conexión entre las neuronas y se transfieren señales de tipo eléctrico o químico

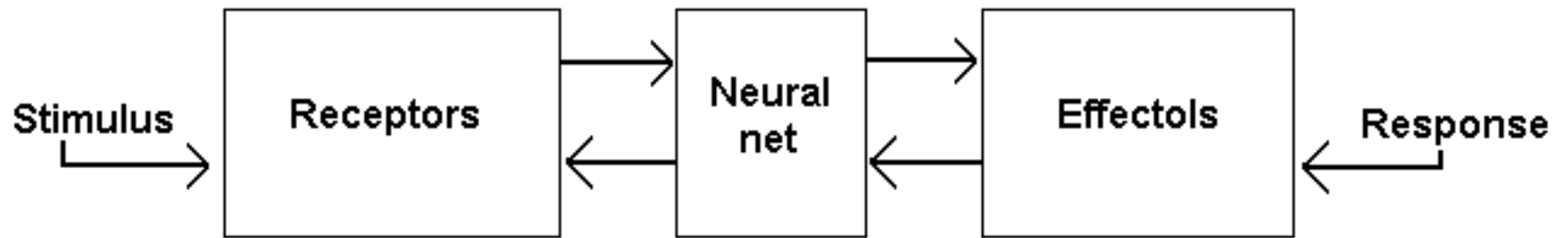
Neuronas



RED NEURONAL BIOLÓGICA

- ***CONJUNTO DE MUCHAS NEURONAS INTERCONECTADAS***
- ***PROCESAMIENTO DE LA INFORMACIÓN BASADA EN LA DESCOMPOSICIÓN DE LA MISMA***

SISTEMA NERVIOSO



RN biológica

- ◆ Se nace con alguna estructura neural
- ◆ Se crean nuevas conexiones y otras se gastan; se desarrollan a través del aprendizaje propio de la etapa de crecimiento.
- ◆ La estructura neural cambia a través de la vida. Esos cambios consisten en reforzamiento o debilitamiento de las junturas sinápticas.
- ◆ Se forman nuevas memorias al modificar o reforzar algunas sinápsis.
- ◆ Por ejemplo, memorizar la cara de una persona que nos presentan, consiste en alterar varias sinápsis.

PROPIEDADES DE LAS REDES NEURONALES

- ***MUCHAS CONEXIONES PARALELAS ENTRE NEURONAS***
- ***MUCHAS CONEXIONES PARALELAS PROVEEN MECANISMOS DE RETROALIMENTACIÓN PARA ELLA U OTRAS***
- ***ALGUNAS NEURONAS PUEDEN EXCITAR UNAS NEURONAS MIENTRAS INHIBEN A OTRAS***
- ***EJECUTAN UN PROGRAMA QUE ES DISTRIBUIDO***
- ***TIENEN PARTES PRE-HECHAS Y OTRAS QUE EVOLUCIONAN***

PROPIEDADES DE LAS REDES NEURONALES

- ***SON SINCRÓNICAS***
- ***EJECUTAN UN PROGRAMA QUE ES
DISTRIBUIDO***

RED NEURONAL

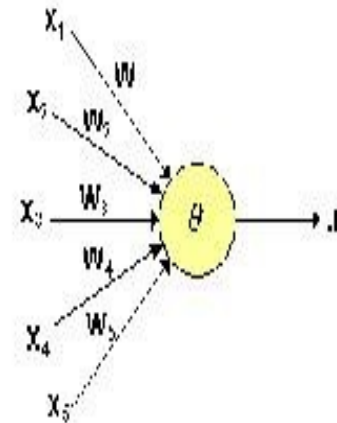
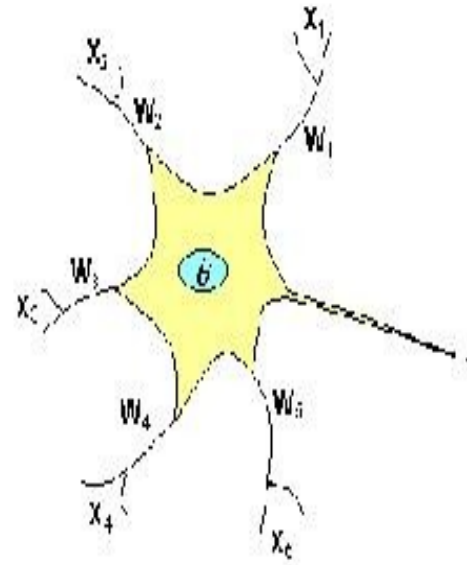
- Capacidad de procesamiento paralelo.
- Capacidad adaptativa.
- Capacidad asociativa.
- Capacidad de auto-organización.
- Capacidad de generalización, clasificación, extracción y optimización.
- Tolerancia a Fallas
- Operación en tiempo real

INTRODUCCIÓN A LAS RNAs

- ***COMO LA RED NEURONAL HUMANA ES DISEÑADA?***
- ***COMO EL CEREBRO PROCESA LA INFORMACIÓN?***
- ***CON QUE ALGORITMOS Y ARITMÉTICA EL CEREBRO CALCULA?***
- ***COMO PUEDE EL CEREBRO IMAGINAR?***
- ***COMO PUEDE EL CEREBRO INVENTAR?***
- ***QUE ES PENSAR?***
- ***QUE ES SENTIR?***

Definiciones de RNA

- ✓ Una nueva forma de computación, inspirada en modelos biológicos.
- ✓ Un modelo matemático compuesto por un gran número de elementos procésales en niveles.
- ✓ Son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica.



Modelo de una RNA vs. el Modelo Biológico

REDES NEURONALES ARTIFICIALES – ANALOGÍA

Neurona Biológica	Neurona Artificial
Señales que llegan a la sinapsis	Entradas a la neurona
Carácter excitador o inhibidor de la sinapsis de entrada	Pesos de entrada
Estimulo total de la neurona	Sumatoria de pesos por entradas
Activación o no de la neurona	Función de activación
Respuesta de la neurona	Función de salida

RNA

	Cerebro	Computador
Velocidad Procesamiento	10^{-2} seg	10^{-9} seg
Tipo procesamiento	Paralelo	Secuencial
Numero procesadores	10^{11}	Pocos
Conexiones	10000 por neurona	Pocas
Almacenamiento conocimiento	Distribuido	Direcciones fijas
Tipo control de proceso	Auto-organizado	centralizado
Tolerancia a Fallas	Amplia	nula

RED NEURONAL (aplicaciones)

- ***Tareas cognitivas (reconocimiento de formas, aprendizaje, guía, sistemas expertos, etc.),***
- ***Control adaptativo (robótica, inspección automática, etc.),***
- ***Procesamiento de conocimiento difuso (sistemas financieros, etc.),***
- ***Optimización, predicción, modelado,***
- ***Procesamiento de señales, de la visión y de la palabra,***
- ***Gestión de la información***

RED NEURONAL (aplicaciones)

- ***Evaluar las probabilidades de formaciones geológicas y petrolíferas.***
- ***Previsión del tiempo.***
- ***Previsión de la evolución de los precios, para valorar los riesgos de los créditos, para interpretar firmas, etc.***
- ***Controlar la producción, que inspeccionan la calidad, etc.***
- ***En el área médica, para diagnosticar y tratar a partir de datos analíticos y/o síntomas,***

NEURONA

- ***UNIDAD FUNDAMENTAL DEL SISTEMA NERVIOSO***
- ***PROCESADOR DE SEÑALES ELÉCTRICAS Y BIOQUÍMICAS***
- ***RECIBE Y COMBINA SEÑALES DESDE MUCHAS NEURONAS***
- ***SINAPSIS: UNIDAD FUNCIONAL QUE INTERRELACIONA LAS NEURONAS***
- ***POTENCIAL POSTSINÁPTICA: PUEDE SER POSITIVO (EXCITACIÓN) O NEGATIVO (INHIBICIÓN)***
- ***NEUROTRANSMISOR: GENERA POLARIZACIÓN PARA LA MEMBRANA POSTSINÁPTICA***

NEURONA

La función de la neurona es transmitir información.

Esa información se transmite en la forma de impulsos nerviosos.

El impulso viaja en una sola dirección: se inicia en las dendritas, se concentra en el soma y pasa a lo largo del axón hacia otra neurona, músculo o glándula.

El impulso nervioso es de naturaleza electroquímica, o sea, que es una corriente eléctrica producida por gradientes de concentraciones de sustancias químicas que tienen cargas eléctricas.

El proceso global de transmisión de un impulso nervioso puede ser dividido en varias fases:

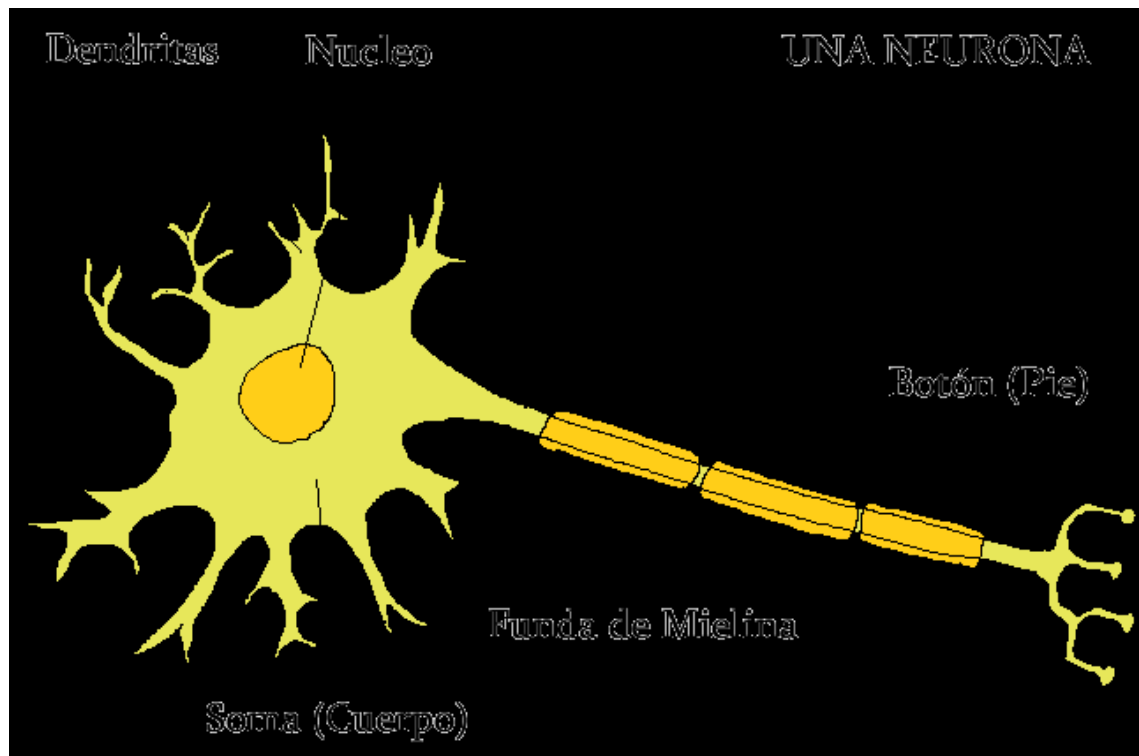
- El potencial de reposo,

- El potencial de acción,

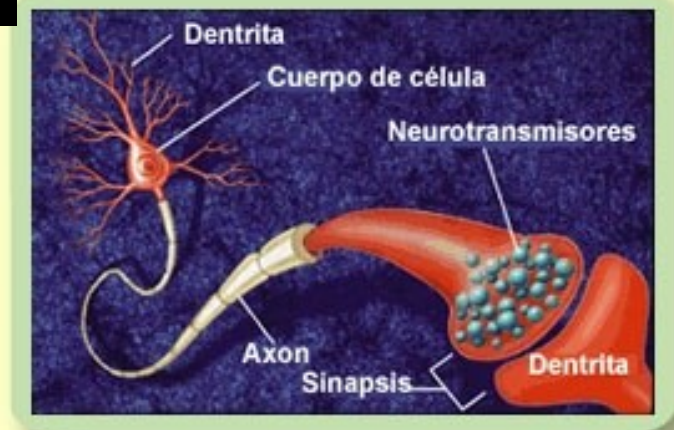
- El desplazamiento del potencial de acción a lo largo del axón y

- La transmisión sináptica.

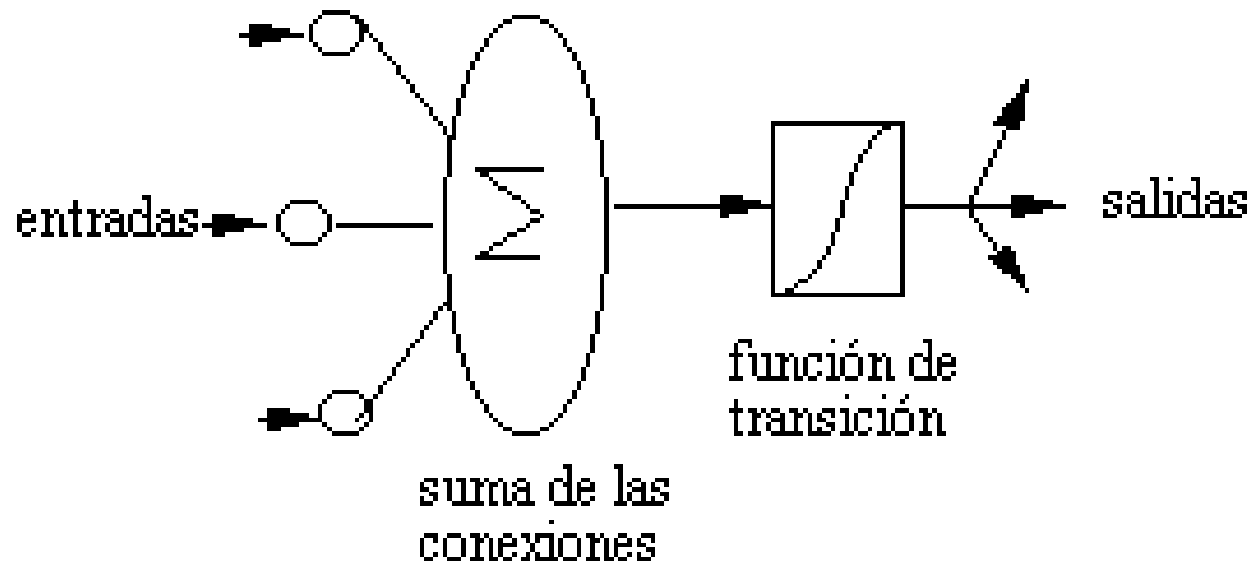
NEURONA



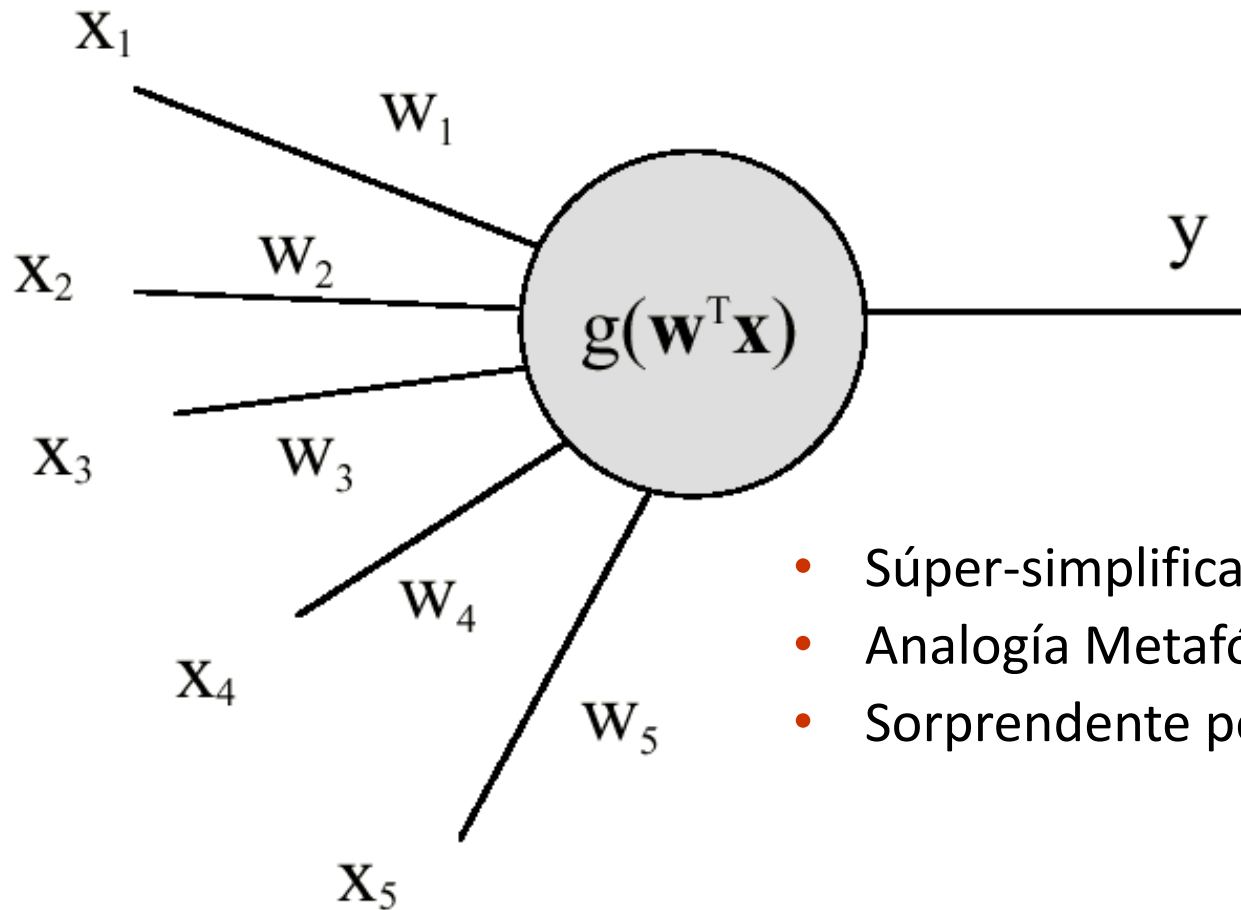
J. AGUILAR



COMO TRABAJA UNA NEURONA



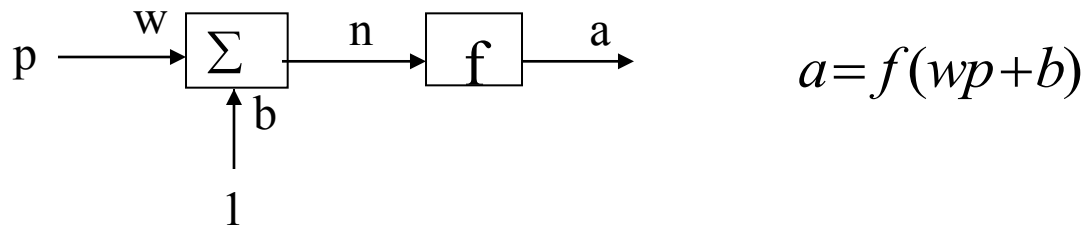
Neuronas Artificiales



- Súper-simplificación
- Analogía Metafórica
- Sorprendente poder de cómputo

Modelo matemático de la neurona

Neurona de una entrada



La entrada escalar p se multiplica por el peso escalar w para formar wp , un término que entra al sumador.

Otra entrada 1 , que se multiplica por un sesgo (offset) b , también va al sumador.

La salida del sumador n , que se conoce como entrada de red va a la función de transferencia f (función de activación), la cual produce la salida escalar a de la neurona.

Correspondencia con la neurona biológica

El peso w se corresponde con el efecto de la sinápsis, el cuerpo de la célula se representa por la sumatoria y la función de transferencia, y la salida a de la neurona representa la señal sobre el axón.

La salida depende de la función de transferencia que se escoja.

El sesgo b es un peso con una entrada constante de valor 1, (se puede omitir).

W y b son los parámetros ajustables de la neurona.

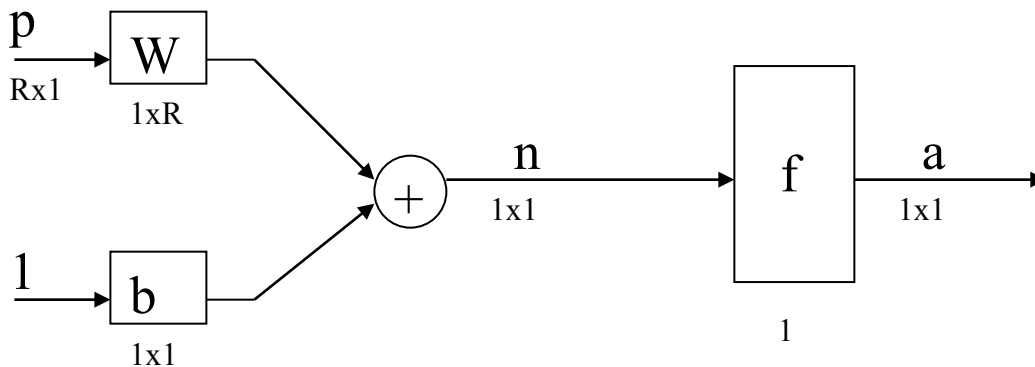
La función de transferencia la escoge el diseñador.

Los parámetros w y b se ajustan de acuerdo a una **regla de aprendizaje** de tal manera que la relación entrada/salida de la neurona debe alcanzar algún objetivo.

Modelo matemático de la neurona

Neurona de múltiples entradas

Típicamente una neurona tiene más de una entrada.



$$n = w_{11}p_1 + w_{12}p_2 + \dots + b$$

$$n = Wp + b$$

La salida de la neurona se escribe ahora como:

$$a = f(Wp + b).$$

El número de entradas a la red depende del problema.

Por ejemplo, si se desea predecir las condiciones para volar las entradas podrían ser la temperatura del aire, la velocidad del viento y la humedad (la red necesitará tres entradas).

COMO TRABAJA UNA NEURONA

- ***El estado interno o de activación de las neuronas:*** Los estados del sistema en un tiempo t se representan por un vector $A(t)$. Los valores de activación pueden ser continuos o discretos, limitados o ilimitados

El funcionamiento de una neurona se caracteriza por dos estados: inhibitorio o inactivo y excitatorio o activo, donde la determinación de este estado se da por medio de una función de transferencia.

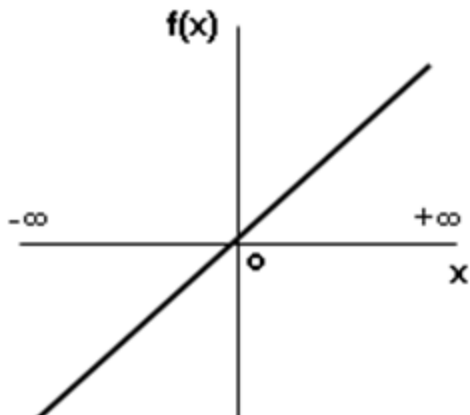
- ***Función de Salida o de Transferencia:*** Asociada con cada unidad hay una función de salida, que transforma el estado actual de activación en una señal de salida. Funciones de transferencia típicas:
 - Función Escalón
 - Función Lineal y Mixta
 - Sigmoidal
 - Función Gaussiana

FUNCIONES DE ACTIVACIÓN

Función identidad o función

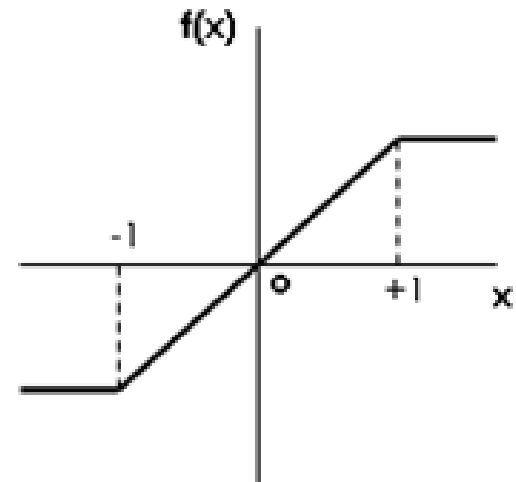
lineal: Es una función continua. La entrada de la neurona es igual a la salida. Su rango es $[-\infty, +\infty]$ y la representación matemática es

$$f(x) = x$$



Función lineal por tramos: Es similar a la función lineal y su rango es $[-1, +1]$.

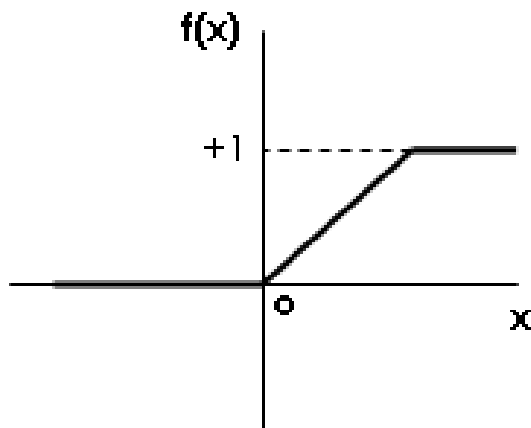
$$f(x) = \begin{cases} -1, & x < -1 \\ x, & -1 \leq x \leq +1 \\ +1, & x > +1 \end{cases}$$



FUNCIONES DE ACTIVACIÓN

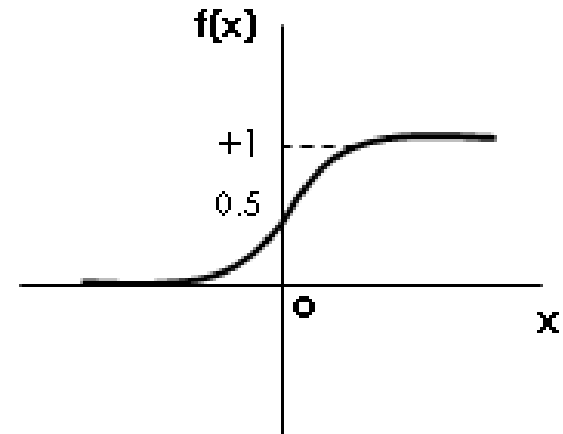
Función escalón: es una función discreta que puede tomar los valores entre 0 y 1. También se le denomina función paso.

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$



Función logística o sigmoideal: es una función continua acotada entre 0 y 1. Este tipo de función es comúnmente utilizada en problemas de predicción.

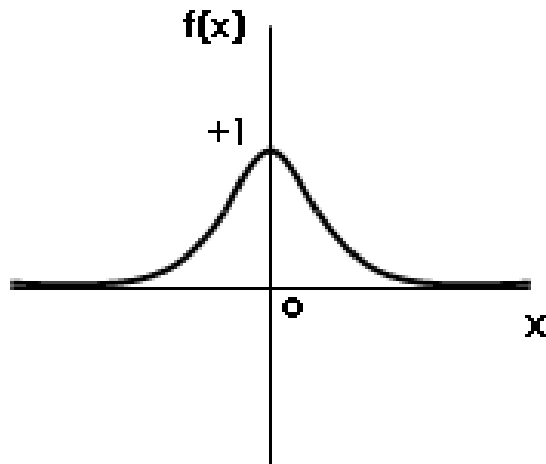
$$f(x) = \frac{1}{1 + e^{-x}}$$



FUNCIONES DE ACTIVACIÓN

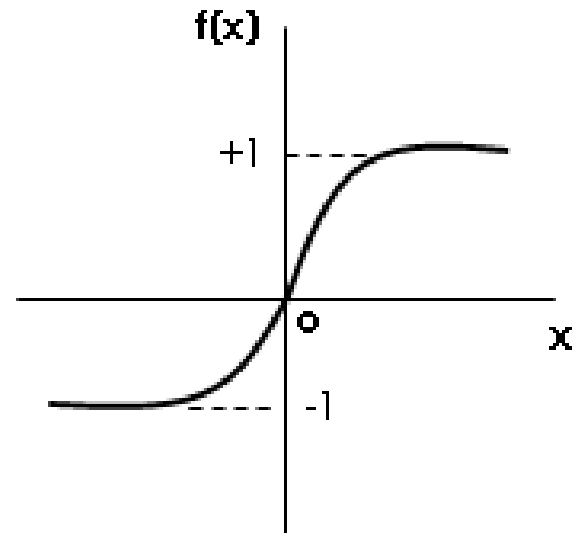
Función gaussiana: es una función continua acotada entre 0 y 1. Se emplea en Redes Neuronales de Funciones de Base Radial.

$$f(x) = Ae^{-Bx^2}$$



Función tangencial hiperbólica: es una función continua acotada similar a la sigmoidea, pero el rango es $[-1, +1]$.

$$f(x) = \tanh(x)$$



Estructura de una RNA

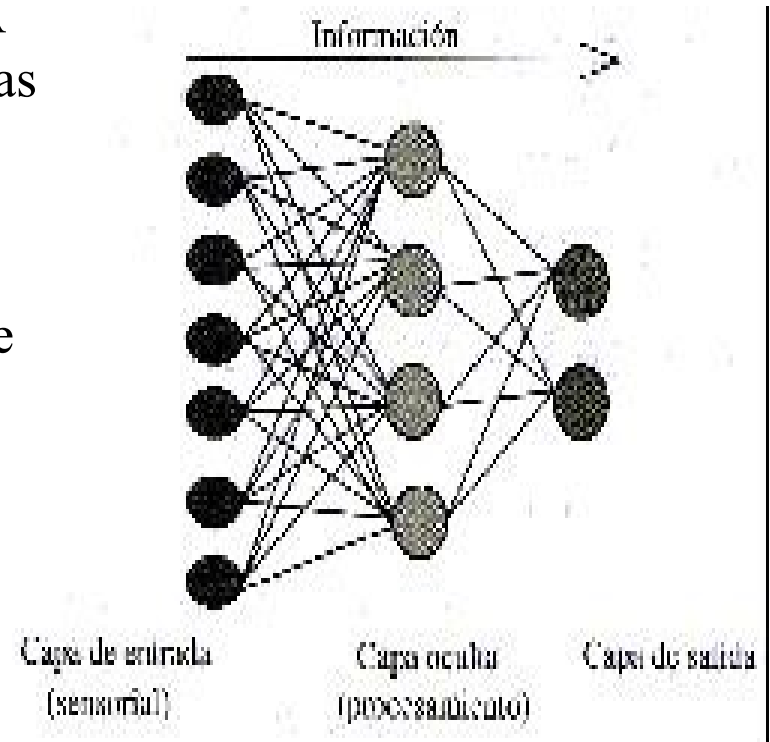
Las conexiones entre las neuronas.

La conectividad entre los nodos de una RNA esta relacionada con la forma en que la salidas de las neuronas están canalizadas para convertirse en entradas de otras neuronas.

Las conexiones que unen a las neuronas que forman una RNA tiene asociado un peso, que es el que hace que la red adquiera conocimiento

Formas de conexión entre neuronas:

- Conexión autorrecurrente
- Propagación hacia delante
- Propagación hacia atrás



Las conexiones con el ambiente.

COMO TRABAJA UNA NEURONA

- ✓ *Tipo de Asociación entre las Informaciones de Entrada y Salida:* Las RNA son sistemas que almacenan cierta información aprendida; esta información se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas de entrada y salida.

AL APLICARLE UNA ENTRADA LA RN RESPONDE CON UNA SALIDA ASOCIADA A DICHA INFORMACIÓN DE ENTRADA

- ✓ Este mecanismo da lugar a dos tipos de RNA:
 - **Red Heteroasociativa:** es aquella que computa cierta función, entre un conjunto de entradas y un conjunto de salidas, correspondiendo a cada posible entrada una determinada salida.
 - **Red Autoasociativa:** es una red cuya misión es reconstruir una determinada información de entrada que se presenta incompleta o distorsionada.

MEMORIAS ASOCIATIVAS

HETEROASOCIATIVAS:

- 1. RED APRENDE PAR DE DATOS (A_i , B_i)***
- 2. SI SE PRESENTA A_i , RN RESPONDE B_i***
- 3. POR LO MENOS DOS CAPAS***
- 4. TODOS LOS TIPOS DE APRENDIZAJE***

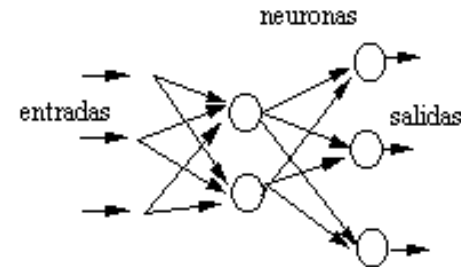
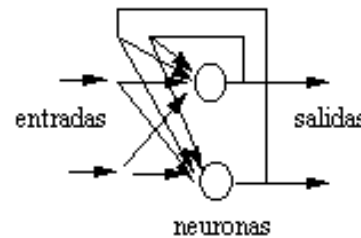
MEMORIAS ASOCIATIVAS

AUTOASOCIATIVA

- 1. RED APRENDE A_i***
- 2. SI SE PRESENTA A_i , RN RESPONDE A_i***
- 3. ASOCIA INFORMACION DE ENTRADA CON LO MAS PARECIDO QUE TENGA GUARDAD***
- 4. PUEDE SER DE 1 CAPA***
- 5. APRENDIZAJE NO SUPERVISADO $\Rightarrow 0$ CONEXIONES RECURRENTES Y/O LATERALES***

COMO TRABAJA UNA NEURONA

- ***La regla de propagación***



- ***La topología o arquitectura de la red:*** La topología de una red neuronal artificial es una estructura que está determinada por el número de capas y la conexión entre ellas.

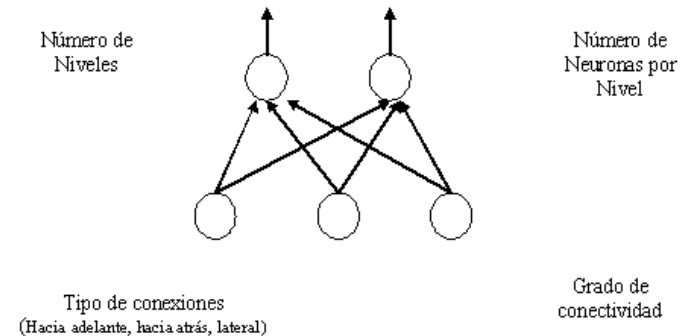
- conexión total (todas las neuronas interconectadas)
- conexión parcial (por ejemplo, las redes de capas).

- ***La regla de aprendizaje:***

El aprendizaje puede ser comprendido como la modificación de comportamiento inducido por la interacción con el entorno y como resultado de experiencias conduce al establecimiento de nuevos modelos de respuesta a estímulos externos.

Topología o arquitectura

- ✓ **Niveles o capas de Neuronas:** En general las neuronas se suelen agrupar en unidades estructurales que se denominan capas. El conjunto de una o mas capas constituye la red neuronal. Se distinguen tres tipos de capas: de entrada, de salida y ocultas.



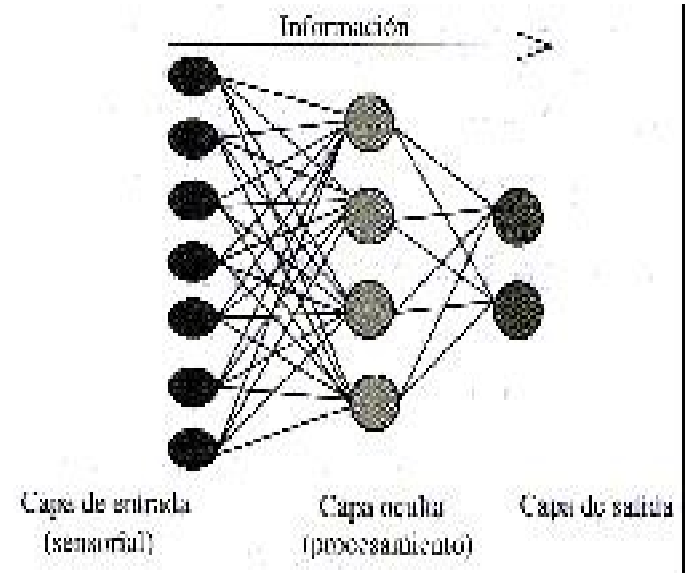
Parámetros de una RNA

Redes monocapa: son redes con una sola capa. Para unirse las neuronas crean conexiones laterales, cruzadas o autorrecurrentes entre las neuronas que pertenecen a la única capa que constituye la red

Redes multicapas: son una generalización de las anteriores donde existe un conjunto de capas intermedias entre la entrada y la salida llamadas capas ocultas. Disponen de conjuntos de neuronas agrupadas en varios niveles o capas.

RNA Multicapa

- ✓ Número de niveles o capas.
- ✓ Número de neuronas por nivel.
- ✓ Patrones de conexión intra-nivel inter-nivel.
- ✓ Flujo de Información.

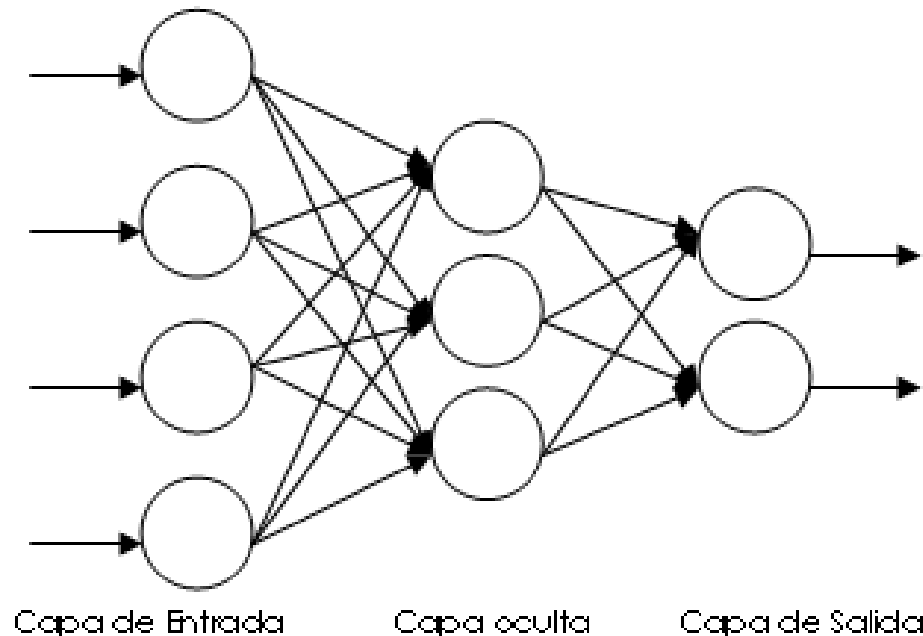


RNA Multicapa

Capa de Entrada: está constituida por los nodos de entrada, que reciben directamente la información de las fuentes externas a la red.

Capas Ocultas: no tienen contacto con el exterior ya que se encuentran ubicadas entre la capa de entrada y la capa de salida. La cantidad de capas ocultas dependerá del problema en estudio y deben especificarse en la arquitectura.

Capa de Salida: está constituida por los nodos que transfieren la información a la salida de la red y de acuerdo al tipo de problema en estudio se determinará el número de neuronas de salida.



Conexión entre neuronas

Propagación hacia adelante: son aquellas donde la propagación de las señales se efectúa en un solo sentido, es decir, de los nodos de salida de una capa hacia los nodos de entrada de la siguiente capa; por ende no hay retroalimentaciones.

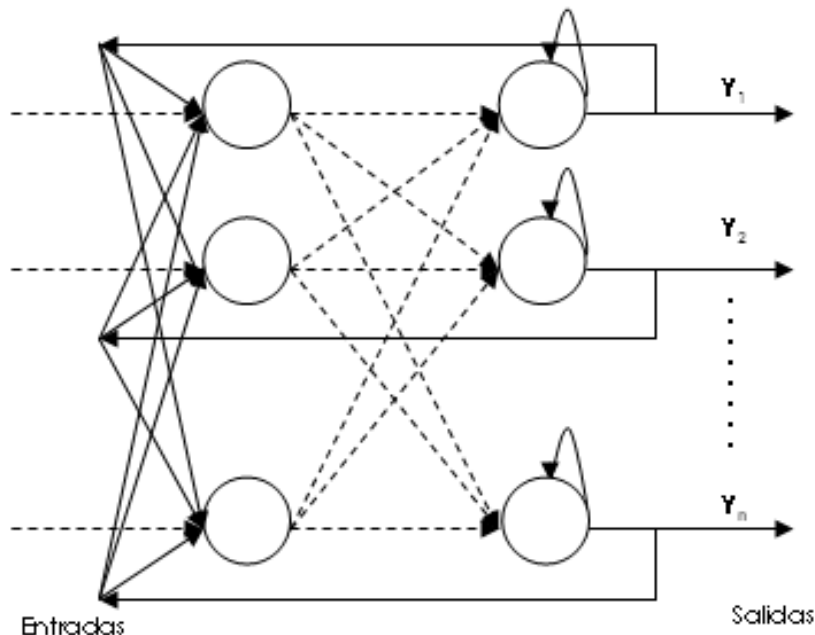
Propagación hacia atrás: Se caracteriza por la existencia de lazos de retroalimentación, es decir, la salida de las neuronas de una capa pueden ser entrada de las neuronas de la misma capa o de capas anteriores. Pueden ser:

Redes recurrentes

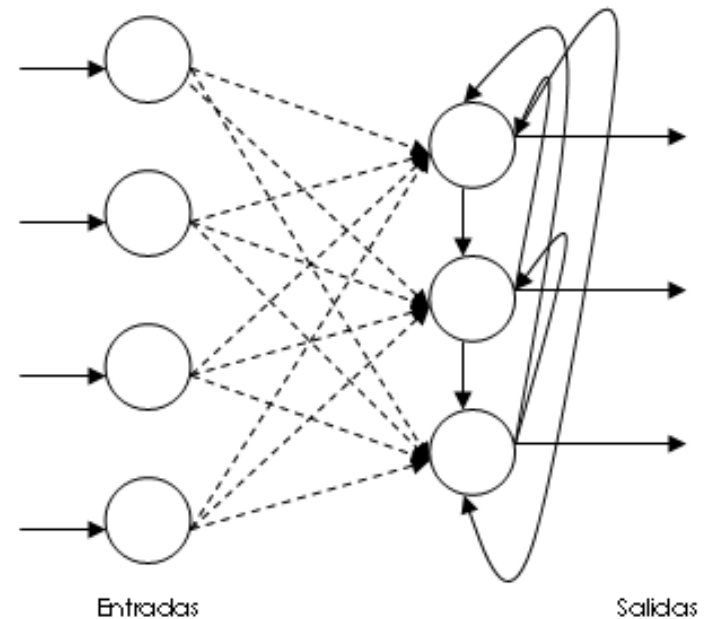
Redes de alimentación lateral

Conexión entre neuronas

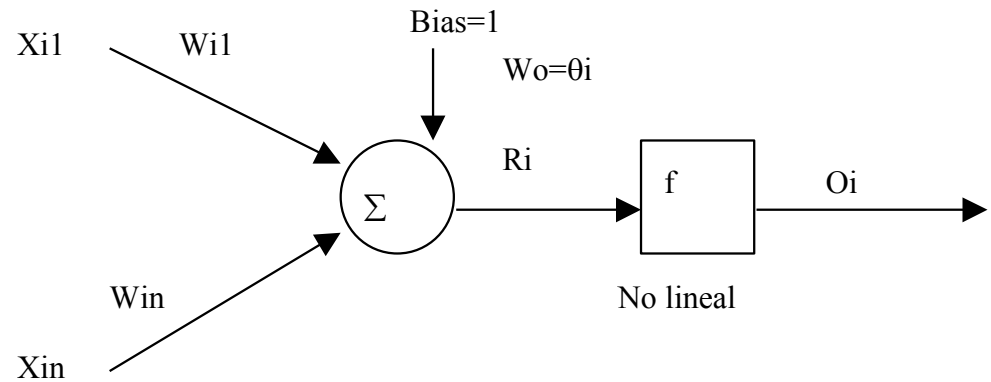
Redes recurrentes: son aquellas redes donde la salida de un nodo es la entrada del mismo nodo, son redes de propagación hacia atrás con lazos cerrados.



Redes de alimentación lateral: se caracterizan porque las salidas de las neuronas pueden ser entradas de neuronas de la misma capa.



MODELO DE McCULLOCH-PITTS



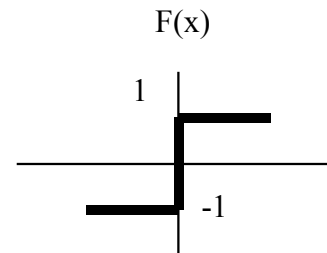
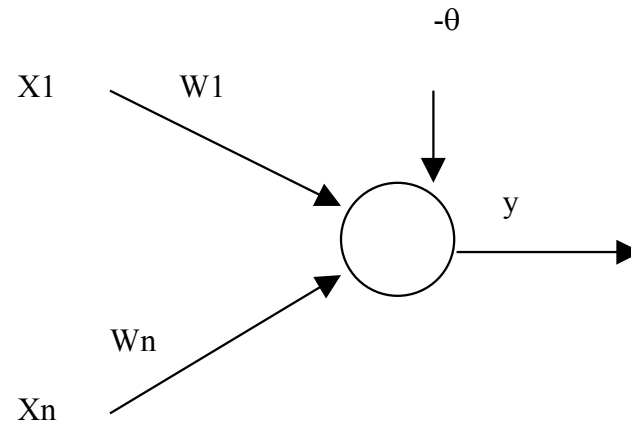
$$O_i = f\left(\sum_{j=1}^N (x_{ij} w_{ij}) - \Theta_i\right)$$

PERCEPTRÓN

- 1^{ER} MODELO DE RED DE NEURONAS ARTIFICIALES (ROSEMBLATT 1958)
- APRENDE PATRONES SENCILLOS (2 CLASES)
- DOS CAPAS Y CAPA DE SALIDA CON 1 NEURONA

✓ Es un modelo unidireccional compuesto por dos capas de neuronas, una de entrada y otra de salida, con 1 neurona.

✓ Se aplica en el calculo del AND, OR, XOR, pero no resuelve el OR-exclusivo



$$Y = F(\sum W_i X_i - \theta)$$

PERCEPTRÓN

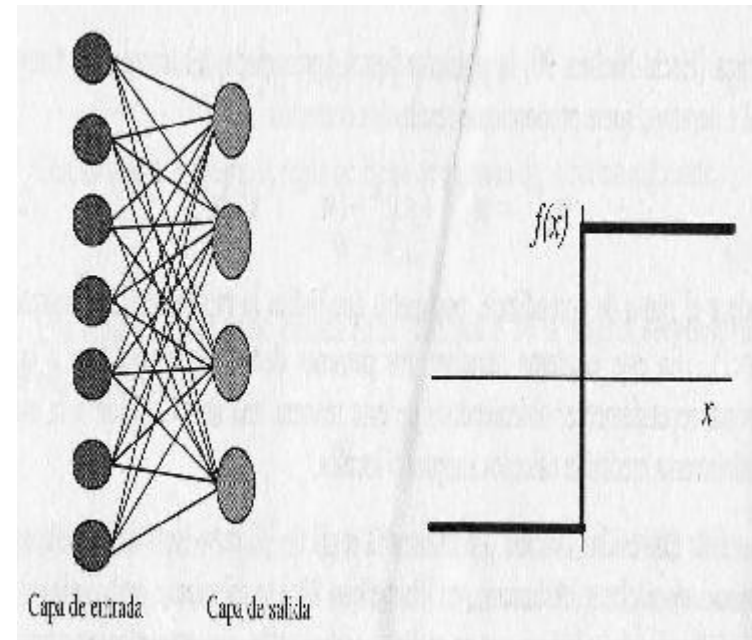
- REGIONES QUE INDICA A QUE PATRÓN PERTENECE CADA CLASE SEPARADAS POR UN HIPERPLANO

=> PATRONES SEPARABLES GEOMÉTRICAMENTE

=> DOS ENTRADAS LINEA RECTA

$$x_2 = w_1 x_1 / w_2 + \theta / w_2$$

=> TRES ENTRADAS PLANO

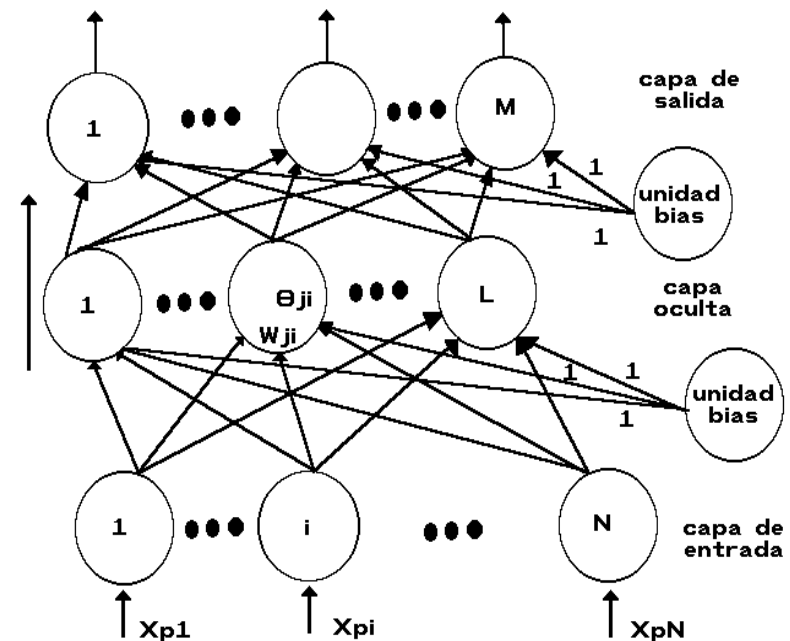


PERCEPTRÓN

- APRENDIZAJE: SUPERVISADO
- ALGORÍTMO:
 1. INICIAR PESO Y UMBRAL
 2. PRESENTAR PAR ENTRADA-SALIDA
 3. CALCULAR SALIDA ACTUAL
 $Y(t)$
 4. ADAPTAR LOS PESOS
 $W_i(t) = W_i(t) + \alpha [d(t) - Y(t)] X_i(t)$
HASTA QUE $d(t) - y(t)^2$ valor pequeño
 5. REGRESAR AL PASO 2

Backpropagation

- ✓ Fue propuesto por Rumelhart, Hinton y Williams, 1986.
- ✓ Aprendizaje pares de Entrada- Salida.
- ✓ Capacidad de autoadaptación pesos capas escondidas
- ✓ Al aplicar un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida.
- ✓ Se aplica en la codificación de información, traducción de texto en lenguaje hablado, clasificación de señales electrocardiográficas y en la comprensión y descompresión de los datos.



Modelo de Backpropagation

**CAPACIDAD DE AUTOADAPTACIÓN PESOS CAPAS ESCONDIDAS
=> GENERALIZACIÓN**

BACKPROPAGATION

- **ALGORÍTMO:**

1. INICIAR PESOS

2. PRESENTAR PAR ENTRADA-SALIDA

3. CALCULAR SALIDA RED
CAPA OCULTA:

$$net_{nj}^k = \sum_{i=1}^N w_{ji}^k x_{pi} + \Theta_j^k \quad ; y_{nj} = f_j^k(net_{nj}^k)$$

n: capa n-esima

p: patrón p

j: neurona j de la capa n

BACKPROPAGATION

CAPA SALIDA:

$$net_{pk}^o = \sum_{i=1}^L w_{ij}^o y_{ji} + \Theta_k^o$$
$$y_{pk} = f_k^o(net_{pk}^o)$$

o: capa de salida

4. CALCULAR ERROR PARA TODAS LAS NEURONAS

CAPA DE SALIDA:

$$\delta_{pk}^o = (d_{pk} - y_{pk}) f_k^o(net_{pk}^o)$$

$$\delta_{pk}^o = (d_{pk} - y_{pk}) \quad f_k^o = 1$$

$$\delta_{pk}^o = (d_{pk} - y_{pk}) * y_{pk} (1 - y_{pk}) \quad f_k^{o'} = f_k^o (1 - f_k^o) = y_{pk} (1 - y_{pk})$$

f_k^{o'} derivable lineal sigmoidal

BACKPROPAGATION

CAPA OCULTA

$$\delta_{ji}^k = f_j^k(\text{net}_{ji}^k) \sum_k \delta_{pk}^o w_{ij}^o$$

$$\delta_{ji}^k = x_{ji}(1 - x_{ji}) \sum_k \delta_{pk}^o w_{ij}^o$$

5. ACTUALIZAR PESOS

CAPA SALIDA

$$w_{ij}^o(t+1) = w_{ij}^o(t) + \Delta w_{ij}^o(t+1) \quad ;$$

$$\Delta w_{ij}^o(t+1) = \alpha \delta_{pk}^o y_{ji}$$

CAPA OCULTA

$$w_{ji}^k(t+1) = w_{ji}^k(t) + \Delta w_{ji}^k(t+1)$$

$$\Delta w_{ji}^k(t+1) = \alpha \delta_{ji}^k x_{pi}$$

BACKPROPAGATION

6. REPETIR PROCESO HASTA QUE

$$E_p = \frac{1}{2} \sum_{k=1}^M e_{pk}^2$$

M: número neuronas salidas
PARA CADA PATRÓN

Aprendizaje

Jose Aguilar
Cemisid, Facultad de Ingeniería
Universidad de los Andes
Mérida, Venezuela
aguilar@ing.ula.ve

Redes Neuronales Artificiales

RNA:

Habilidad de aprender del medio ambiente y mejorar su desempeño por medio del aprendizaje.

Aprendizaje:

Proceso por el cual los parámetros (pesos sinápticos) de una RNA se adaptan por estimulación del medio ambiente.

Algoritmo de Aprendizaje:

Conjunto bien definido de reglas para actualizar los pesos sinápticos.

Paradigma de Aprendizaje:

Modelo del medio ambiente en el cual la RNA opera.

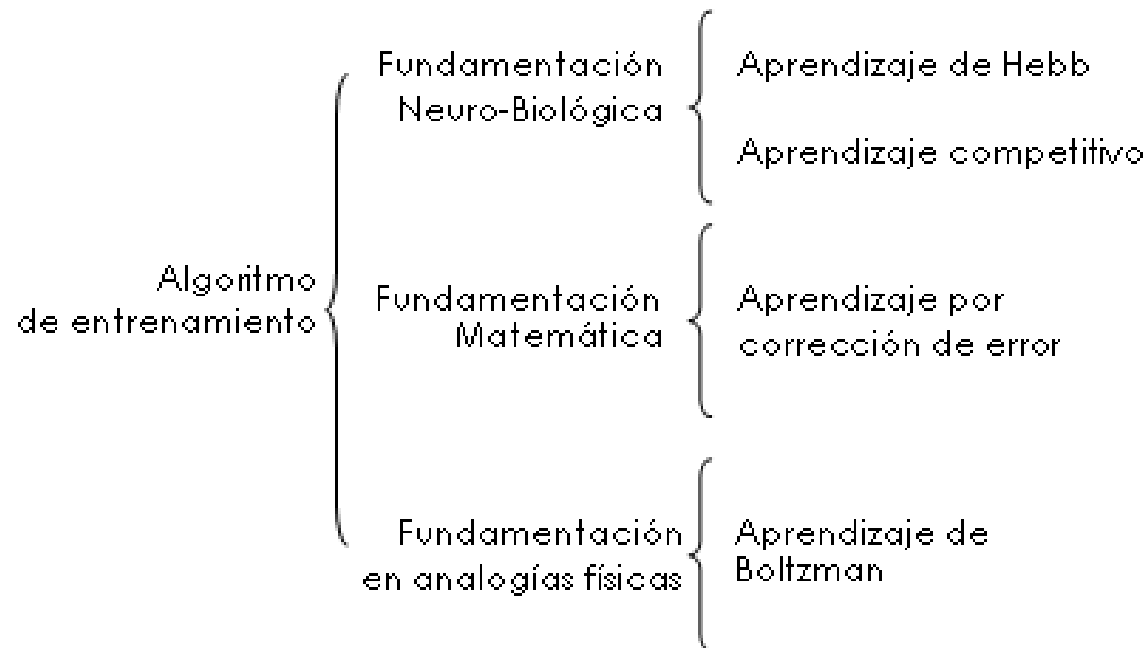
APRENDIZAJE

- ***REDES NEURALES:***
 - ***CAPACIDAD PARA APRENDER***
 - ***OPTIMIZA SU RENDIMIENTO ASI***

APRENDIZAJE: MODIFICAR PESOS DE LAS CONEXIONES DE LAS NEURONAS (CREAR, DESTRUIR, MODIFICAR)

$$W_{ij} = W_{ij} + \Delta W_{ij}$$

ENTRENAMIENTO O APRENDIZAJE



Clasificación de los algoritmos de entrenamiento basados en su fundamentación conceptual

Modelos Neuronales

Clasificación por tipo de aprendizaje y arquitectura

Híbridos

Supervisados

Realimentados

Unidireccionales

No supervisados

Realimentados

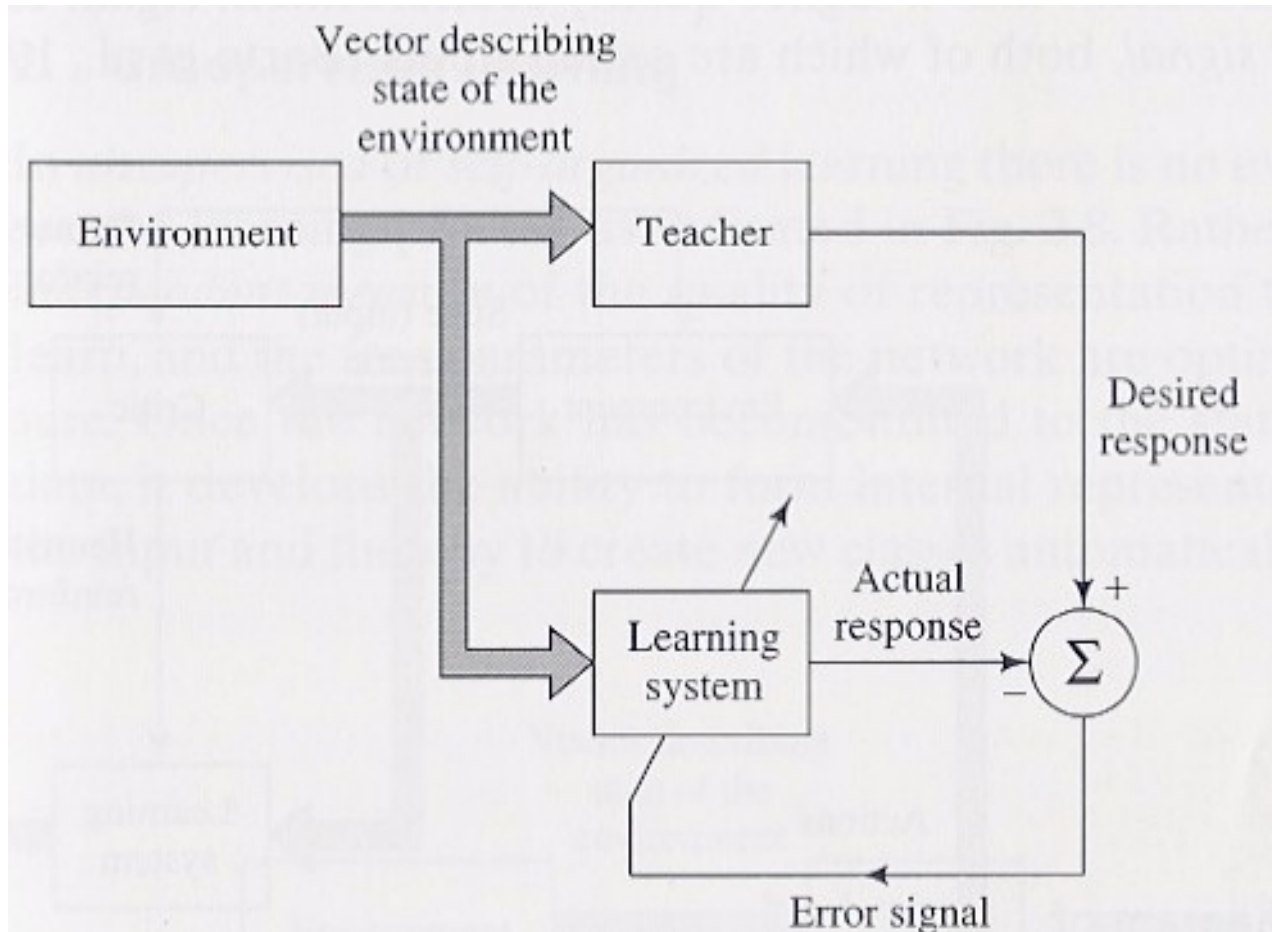
Unidireccionales

Reforzados

Características de una RNA

- ✓ *Redes con Aprendizaje Supervisado:* Se realiza mediante un entrenamiento controlado por un agente externo que determina la respuesta que debería generar la red a partir de una entrada determinada.
 - *Aprendizaje por corrección de error:* Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red.
 - *Aprendizaje por refuerzo:* La función del supervisor es indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito=+1 o fracaso=-1)
 - *Aprendizaje estocástico:* Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red.

Paradigmas de Aprendizaje Supervizado



SUPERVISADO

A. CONTROLADO POR SUPERVISOR

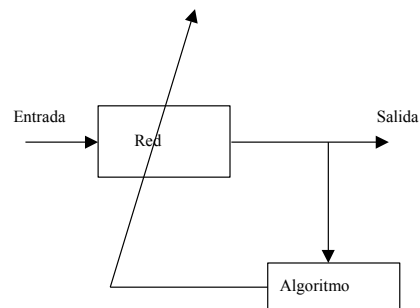
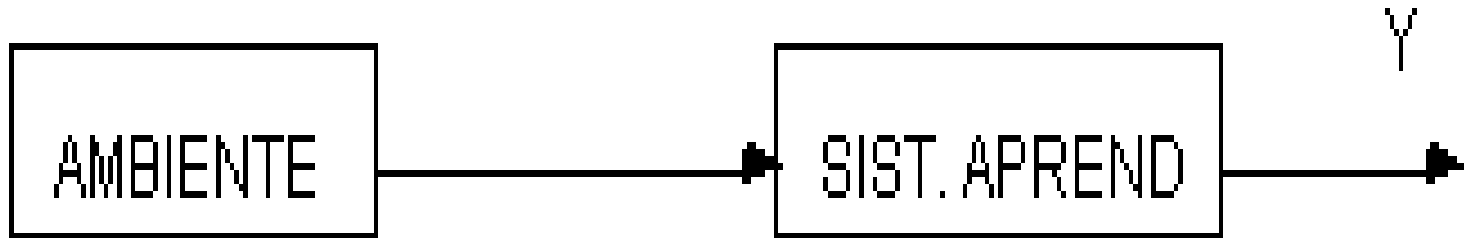
B. SALIDA RED DEBE CONCORDAR CON LA DESEADA

C. SE DAN DATOS DE ENTRADA Y SALIDA

NO SUPERVISADO (AUTOORGANIZADO)

- A. NO RECIBE INFORMACIÓN DE SU ENTORNO***
- B. CON LOS DATOS SE BUSCAN CORRELACIONES
PARA VER QUE TAN BIEN ESTÁ RINDIENDO LA
RED***
- C. RENDIMIENTO SE MIDE SEGÚN COMO UNO
INTÉRPRETE RESULTADOS***
- D. MAPAS TOPOGRÁFICOS***

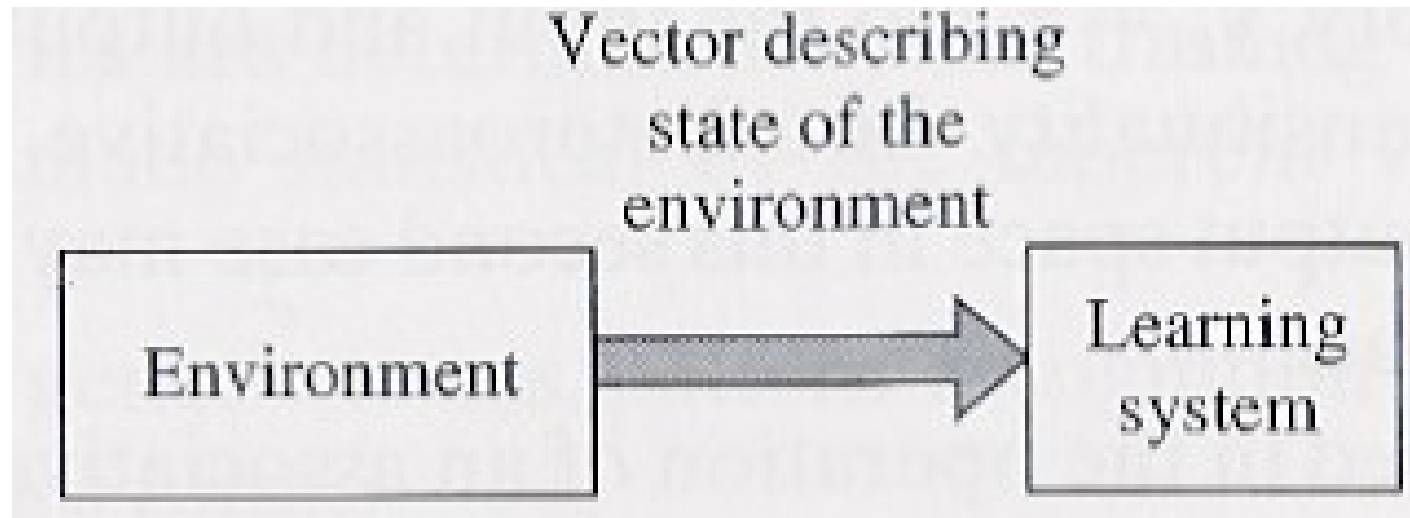
NO SUPERVISADO (AUTOORGANIZADO)



Paradigmas de Aprendizaje

No Supervizado

- Auto-organizado
- Medida de calidad de representación, independiente del contexto
- Los parámetros son optimizados con respecto a esa medida
- Clasificación

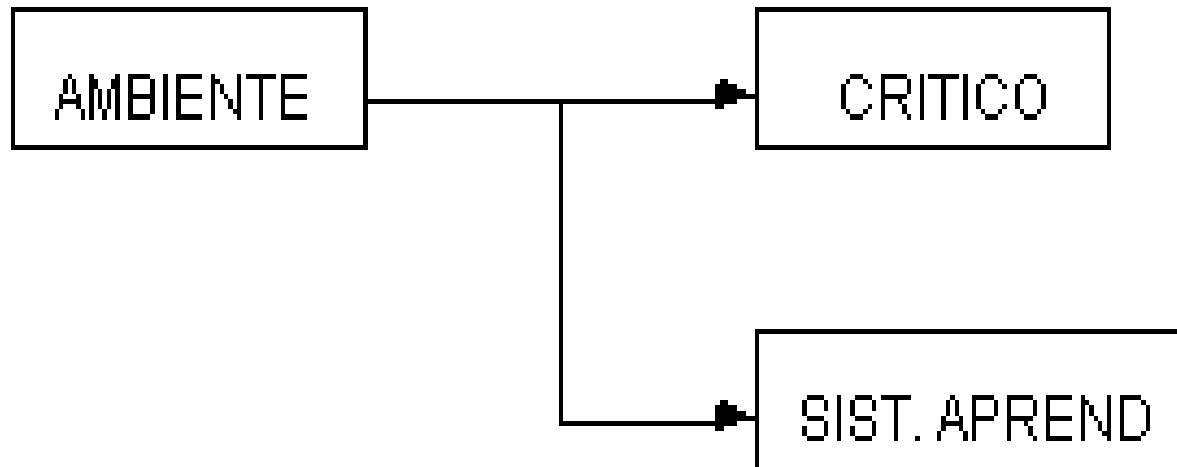


REFORZADO

***A. SUPERVISOR INDICA SI SALIDA SE AJUSTA
A LO DESEADO O NO***

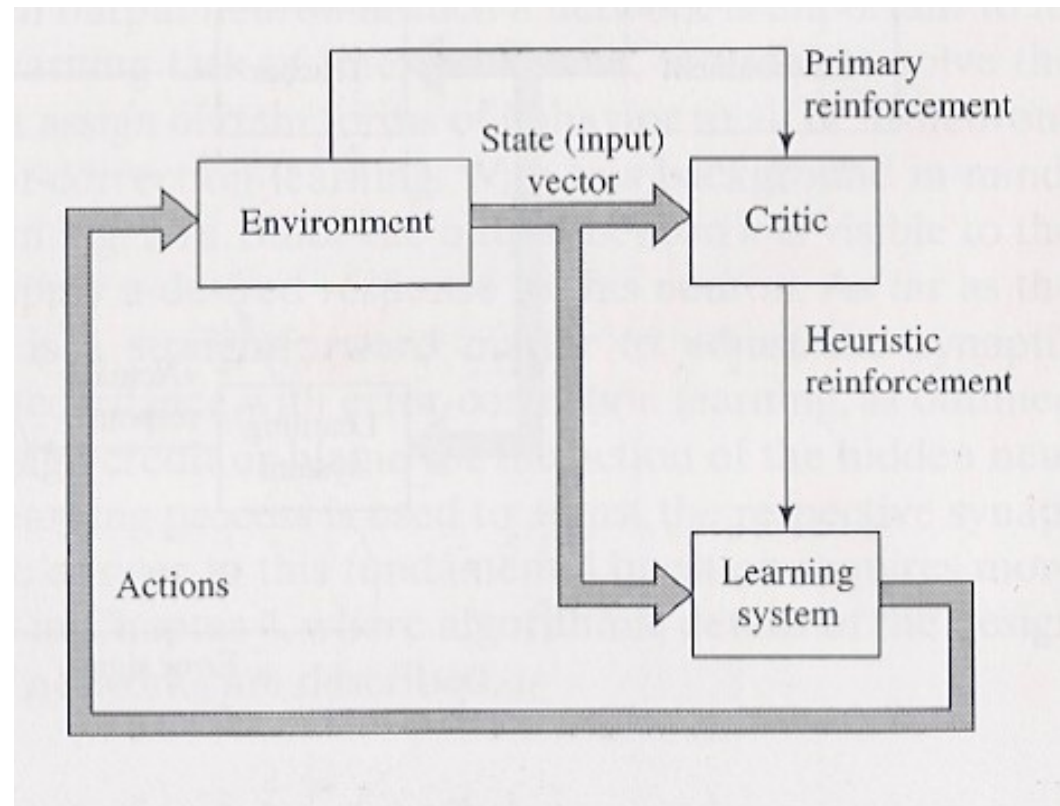
***B. SUPERVISOR HACE PAPEL DE CRÍTICO MÁS
QUE DE MAESTRO***

REFORZADO



Paradigmas de Aprendizaje Por Refuerzos

- Minimizar un índice escalar de aptitud
- Refuerzo retardado
- Asignación de Crédito y Culpa a acciones
- Aprende a realizar tareas basado solamente en el resultado de sus experiencias



Estrategias de Aprendizaje

CORRECCIÓN DE ERROR

SUPERVISADO

CONOCIDO TAMBIEN COMO **DESCENSO DE GRADIENTE**

$$E_k(t) = D_k(t) - Y_k(t)$$

D_k : respuesta deseada

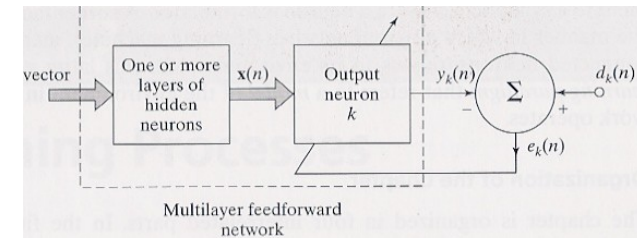
Y_k : respuesta neurona k

X_k : entrada neurona k

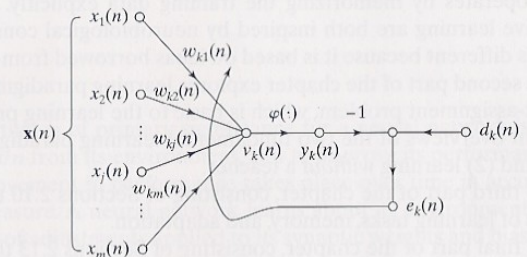
$$Y_k = F(X_k)$$

α : tasa de aprendizaje

$$\Delta W_{ij}(t) = \alpha E_i(t) X_j(t)$$



(a) Block diagram of a neural network, highlighting the only neuron in the output layer



(b) Signal-flow graph of output neuron

REGLA DE WIDROW-HOLF

$$\text{ERROR GLOBAL} = \frac{1}{2P} \sum_{K=1}^P \sum_{J=1}^N \left(Y_J^K - d_J^K \right)^2$$

N: Número de Neuronas

P: Número de Información a Aprender

$$\Delta W_{ji} = \alpha \partial \text{ERRORGLOBAL} / \partial W_{ji}$$

CORRECCIÓN DE ERROR

" Regla Delta":

$$E_{global} = E/2 \sum p \sum n (E_k^2(t))$$

p: No ejemplos n: No neuronas

$$\Delta W_{ij}(t) = \alpha dE_{global}/dW_{ij}$$

CORRECCIÓN DE ERROR

RETROPROPAGACIÓN DEL ERROR

- 1. CALCULAR EDO. DE LA RED (Y_i)***
- 2. CALCULAR ERROR (1)***
- 3. MINIMIZAR ERROR (2)***
- 4. AJUSTAR PESOS***

$$W_{ij} = W_{ij} + \Delta W_{ij}$$

HEBBIANO

A. MÁS VIEJO (no supervisado)

***B. DOS O MAS NEURONAS ACTIVADAS
SIMULTANEAMENTE***

=> REFORZAR LA CONEXIÓN ENTRE ELLAS

$$\Delta W_{ij} = \alpha Y_i Y_j$$

Algoritmos de Aprendizaje Hebbian

1. Si dos neuronas vecinas se activan simultáneamente, el peso de su conexión se incrementa.
2. Si dos neuronas vecinas se activan asíncronamente, la conexión se debilita o elimina.

La conexión se llama sinapsis hebbiana

Algoritmos de Aprendizaje Hebbian

W_{kj} : peso sináptico

x_j : señal presináptica

y_k : señal postsináptica

$$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) \quad \text{— hipótesis de Hebb}$$

Algoritmos de Aprendizaje Basado en Memoria

Almacenar ejemplos entrada – salida

$$\{(x_i, d_i)\} \quad i=1, \dots, N$$

Ejemplo: Clasificación binaria

Clases: C_1 y C_2

x_{test} (nuevo) es clasificado examinando su
vecindad

Algoritmos de Aprendizaje Basado en Memoria

Componentes:

- Criterio para definir vecindad
- Regla de aprendizaje

Ejemplo: Vecino más Cercano (Nearest Neighbor)

x^* e $\{x_1, x_2, \dots, x_N\}$ es el vmc de x_{test} si

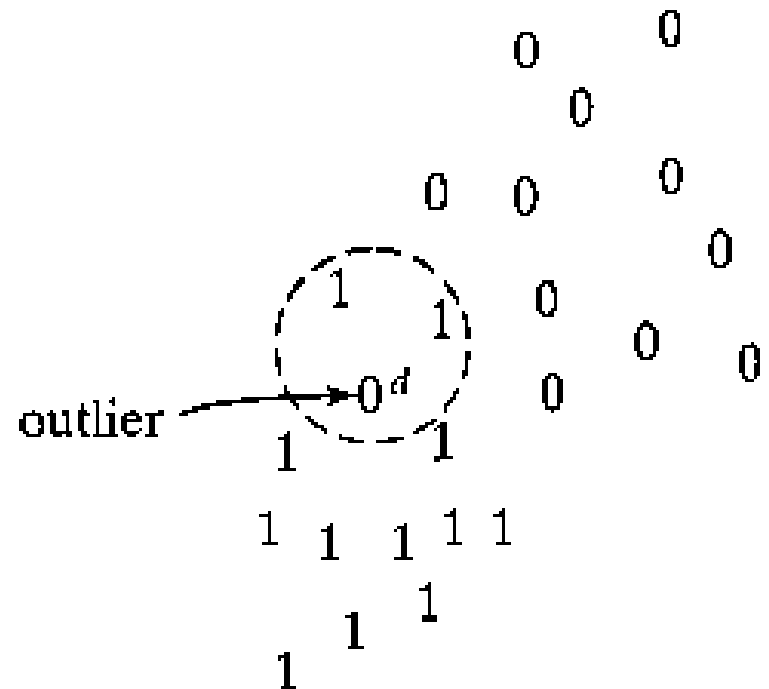
$$\text{Min}_i d(x_i, x_{\text{test}}) = d(x^*, x_{\text{test}})$$

Donde $d(x_i, x_{\text{test}})$ es la distancia Euclideana

Algoritmos de Aprendizaje Basado en Memoria

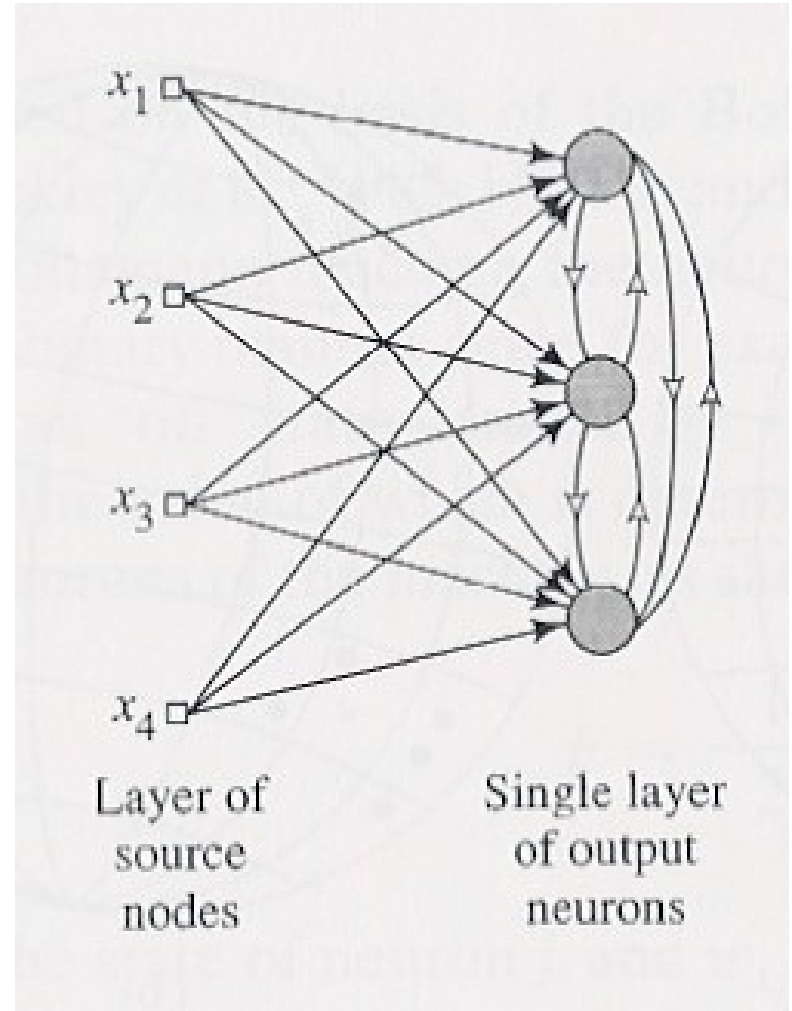
k Vecinos más cercanos:

- Identificar los k vecinos mas cercanos a x_{test}
- Asignar a x_{test} la clase más común en sus k vecinos más cercanos



Algoritmos de Aprendizaje Competitivo

- Las neuronas de salida compiten para activarse.
- Solo una neurona de salida estará activa a la vez.
- Usadas en clasificación.



COMPETITIVO

A. NEURONAS COMPITEN (COOPERAN PARA HACER UNA TAREA)

B. APRENDIZAJE

=> NEURONAS QUE COMPITEN SE INHIBEN ,

NEURONAS QUE COOPERAN SE EXCITAN

=> W_{ij} REFORZADO SI NEURONA j NO ES EXCITADA POR i Y VICEVERSA

$$\Delta W_{ij} = \begin{cases} \alpha (X_i - W_{ij}) & \text{SI NEURONA } j \text{ GANA} \\ 0 & \text{SI NEURONA } j \text{ PIERDE} \end{cases}$$

Algoritmos de Aprendizaje Competitivo

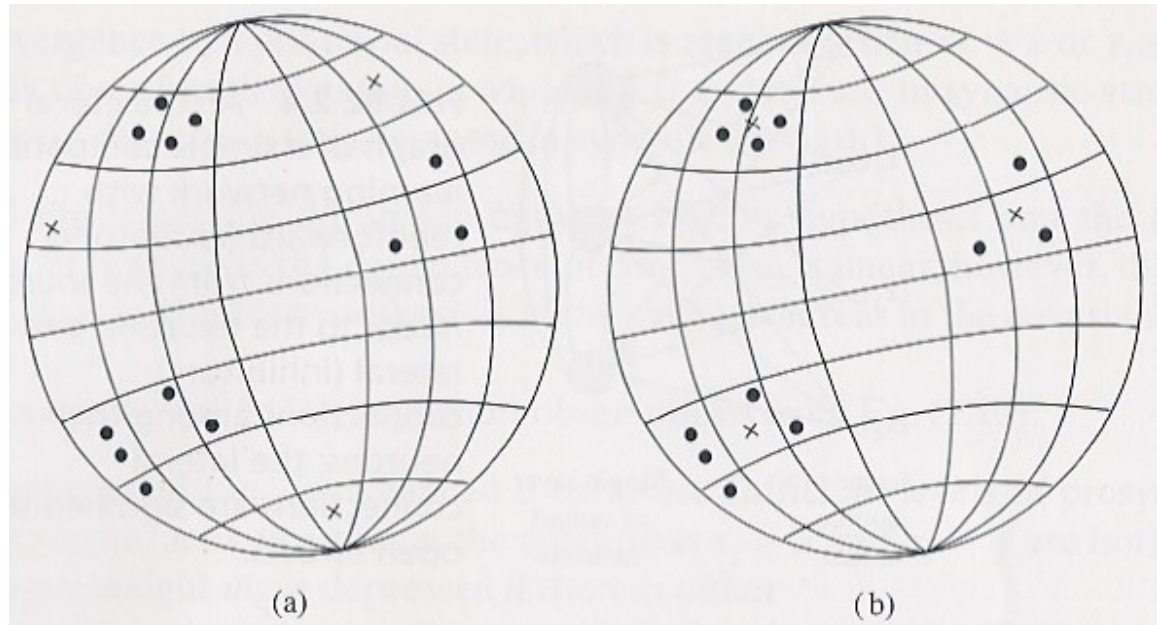
$$y_k = 1 \text{ si } v_k > v_j \quad \forall j \neq k$$

$y_k = 0$ otro caso

$$\sum_i w_{kj} = 1 \quad \forall k$$

$$\Delta w_{kj} = \eta (x_j - w_{kj})$$

$$\Delta w_{kj} = 0$$



REFORZADO

Señal de refuerzo

$$W_{ji}(t) = W_{ji}(t-1) + \alpha (r - \sigma_j) e_{ji}$$

donde

$$e_{ji} = \partial n(g_i) / \partial W_{ji}$$

$$g_i = \Pr(y_j = b_{kj} / W_{ji}, A_k)$$

Modelos Neuronales

BUSQUEDA DE LA MEJOR EMULACION CEREBRAL

PROGRAMA DISTRIBUIDO DE ALMACENAMIENTO DE SINAPSIS

ALGUNOS MODELOS

PERCEPTRON

BOLTZMAN

HOPFIELD

KOHONEN

ART (ADAPTIVE RESONANCE THEORY)

ADALINE

RBF (RADIAL BASIC FUNCTION)

MODELO NEURONAL ALEATORIO

Y MUCHOS MAS

Computación Evolutiva

Jose Aguilar

Cemisid, Facultad de Ingeniería

Universidad de los Andes

Mérida, Venezuela

aguilar@ula.ve

COMPUTACION EVOLUTIVA

- OPTIMIZACION
- ADAPTACION
- MAQUINAS INTELIGENTES
- BIOLOGIA
- CONSTRUCCION DE PATRONES

COMPUTACION EVOLUTIVA

- ENFOQUES ALTERNATIVOS DE BUSQUEDA Y APRENDIZAJE BASADOS EN MODELOS COMPUTACIONALES DE PROCESOS EVOLUTIVOS
- *IDEA:* BUSQUEDA ESTOCASTICA EVOLUCIONANDO A UN CONJUNTO DE ESTRUCTURAS Y SELECCIONANDO DE MODO ITERATIVO A LAS MAS APTAS

FINALIDAD: SUPERVIVENCIA DEL MAS APTO

MODO: ADAPTACION AL ENTORNO

COMPUTACION EVOLUTIVA

- Los esfuerzos de la CE han consistido en relacionar algunos conceptos de la teoría de la evolución con un conjunto de técnicas computacionales
- El principal aporte de la CE ha sido el uso de mecanismos de selección de soluciones potenciales y de construcción de nuevas candidatas por recombinación de características de otras ya existentes.

COMPUTACION EVOLUTIVA

- EMULACION DE PROCESOS EVOLUTIVOS:
 - POBLACION DE POSIBLES SOLUCIONES => INDIVIDUOS
 - PROCESO DE SELECCIÓN => APTITUD DE LOS INDIVIDUOS
 - PROCESO DE TRANSFORMACION => GENERACION DE NUEVOS INDIVIDUOS

COMPUTACION EVOLUTIVA

- FENOMENOS NATURALES SIMULADOS:
 - HERENCIA GENETICA
 - LUCHA POR LA SUPERVIVENCIA
- INSPIRACIONES TEORICAS:
 - EVOLUCION DE DARWIN
 - SELECCIÓN DE WEISMANN
 - GENETICA DE MENDELL

Aspectos generales de la computación evolutiva

El propósito genérico de los algoritmos en la CE es guiar una búsqueda estocástica haciendo evolucionar un conjunto de estructuras, seleccionando de modo iterativo las mas adecuadas.

Características de la CE

- Se trabaja con la codificación de un conjunto de parámetros (o un subconjunto de ellos) y no con los parámetros en si.
- Se usa una función objetivo o función de evaluación para definir lo que se desea optimizar. Esta función permite evaluar la calidad de los individuos en cada generación => función de aptitud

Características de la CE

- Se trabaja con una población del problema, no con un punto simple.
- Se usan reglas de transición probabilísticas, no reglas deterministas

COMPUTACION EVOLUTIVA

- OBJETIVOS CONFLICTIVOS QUE SE SIGUEN:
 - EXPLOTAR LAS MEJORES SOLUCIONES
 - EXPLORAR EL ESPACIO DE BUSQUEDA
- PARADIGMAS:
 - ALGORITMOS GENETICOS (HOLLAND)
 - PROGRAMAS EVOLUTIVOS (MICHALEWICZ)
 - PROGRAMACION GENETICA (KOZA)
 - ESTRATEGIAS EVOLUTIVAS (RECHENBERG SCHWEFEL)
 - PROGRAMACION EVOLUTIVA (FOGEL)

COMPUTACION EVOLUTIVA

- NIVEL DE ABSTRACCION:
 - GENOTIPO => ESTRUCTURAS SON CROMOSOMAS Y GENES. OPERADORES GENETICOS CAMBIAN ESTRUCTURAS (AG)
 - FENOTIPO => ESTRUCTURAS SON COMPORTAMIENTOS. OPERADOR MUTACION CAMBIA ESTRUCTURA (EE,PE).



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

Algoritmos Evolutivos

COMPUTACION EVOLUTIVA

- MACROALGORITMO:
 - 1.- POBLACION INICIAL
 - 2.- EVALUACION DE LOS INDIVIDUOS
 - 3.- REPRODUCCION INICIAL
 - 4.- REEMPLAZO
 - 5.- CONDICION DE FINALIZACION O REGRESA A PASO 2

COMPUTACION EVOLUTIVA

- ASPECTOS DE IMPLEMENTACION:
 - REPRESENTACION DE LOS INDIVIDUOS
 - MEDIDAS DE ADAPTACION (FUNCION ADAPTATIVA)
 - MECANISMOS DE SELECCIÓN Y REMPLAZO
 - INTERPRETACION DE LOS RESULTADOS
 - PARAMETROS Y VARIABLES QUE CONTROLAN EL PROCESO
 - OPERADORES GENETICOS A UTILIZAR

Representación de los Individuos

- Generalidades
- Tipos
 - Cadenas binarias $\Rightarrow 2^l$ soluciones en vector de longitud l
 - Vector de valores reales: incluye vector de desviacion standard
 - Elementos del problema
 - Estados Finitos
 - Arboles

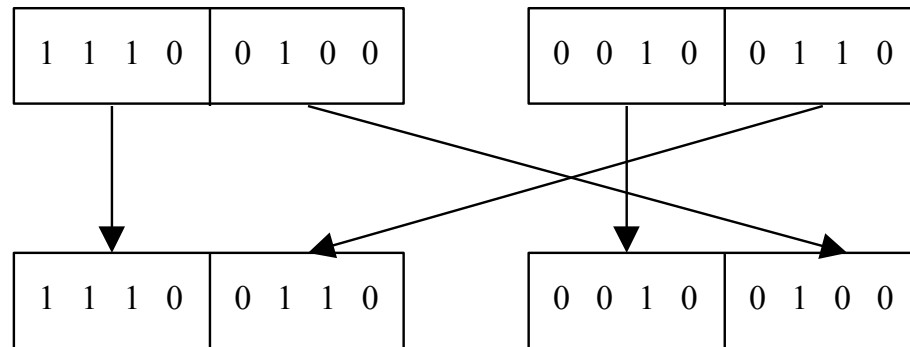
Operadores Genéticos

- Mecanismos de transformación que se utilizan para recorrer el espacio de las soluciones de un problema.
realizan los cambios de la población durante la ejecución de un algoritmo evolutivo.
- En general, se basan en los operadores genéticos biológicos.
- Operadores genéticos clásicos:
 - Mutación
 - Cruce

Operador Cruce

- Operador, normalmente binario,
- Permite representar el proceso de apareamiento natural
- Intercambia pedazos de informacion entre diversos individuos para producir dos nuevos individuos.
- El punto de corte es escogido aleatoriamente.

Operador Cruce

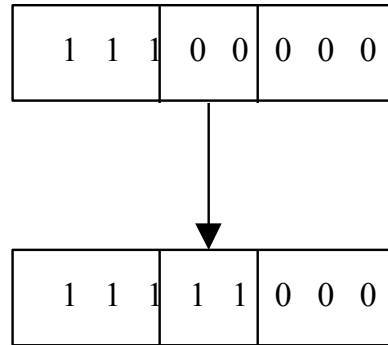


- Tipos:
 - Cruce multipunto
 - Cruce segmentado
 - Cruce uniforme

Operador de Mutación

- Operador unario que simula el proceso evolutivo que ocurre en los individuos cuando cambian su estructura genética.
- Se seleccionan aleatoriamente componentes (genes) de un individuo para ser modificados, también de forma aleatoria.
- A través del operador de mutación se puedan producir cambios en la estructura de un individuo

Operador de Mutación



- Tipos

- Mutaciones sobre genes
- Mutaciones no estacionarias
- Mutaciones no uniformes

Operadores Avanzados

- Dominación
- Segregación
- Inversión
- Duplicación

Mecanismos de funcionamiento de los algoritmos evolutivos

- Estrategias de selección de los individuos susceptibles a reproducirse.
- Estrategia de apareamiento de los individuos en la fase de reproducción.
- Estrategia de reemplazo.

Mecanismos de Selección

- Políticas empleadas para la conformación de los progenitores.
- Tipos:
 - elitesca
 - aleatoria
 - Por sorteo
 - universal (por ruleta)
 - Por torneos

Muestreo por sorteo

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j}$$

- Se calculan las probabilidades acumulativas (q_i):
 - $q_0 = 0$.
 - $q_i = q_{i-1} + p_i$.
- Se genera un número aleatorio α entre 0 y 1.
- Se elige el individuo i que cumpla con la condición:

$$q_{i-1} < \alpha < q_i.$$

Muestreo universal

Se genera un numero aleatorio α y se utiliza este para crear los números:

$$\alpha_i = \frac{\alpha + i - 1}{k} \quad \forall i = 1, k$$

Una vez obtenidos estos números, se aplica k-veces el paso 3 del muestreo por sorteo.

Muestreo por torneos

Se toman al azar m individuos de la población, y el mejor de estos m individuos formara parte de la muestra.

Mecanismos de Reemplazo

- Tipos:
 - Padre por hijo, o Reemplazo directo
 - Por similitud
 - Por inserción (,), p.e., peores por mejores
 - Por inclusión (+)

Estrategia de Apareamiento

- Que individuos con que individuos aparear en la fase de reproduccion
- Tipos:
 - Aleatorio,
 - Autofertilizacion,
 - etc.

ALGUNOS PARAMETROS Y VARIABLES QUE CONTROLAN EL PROCESO

- Probabilidades de uso de los operadores
- Criterios de parada
 - Por ejemplo, Número de generaciones
- Tamaño y diversidad de la población

Revisión Procedimiento general de los algoritmos evolutivos

- Generar una población inicial.
- Evaluar los individuos.
- **Seleccionar**, a través de algún mecanismo, ciertos individuos de la población.
- **Modificar** los genes de los padres seleccionados usando los operadores genéticos.
- Evaluar los nuevos individuos.
- Generar una nueva población con la existente y los individuos generados en el paso 4.
- Verificar el **criterio de convergencia**, o regresar al paso 3.

Aplicaciones

- **Planeación:** enrutamiento, planificación, enpaquetamiento
- **Diseño:** sistemas digitales y electronicos, redes neuronales, reactores, estructuras, aviones, compt. paralela
- **Simulación e Identificación:** modelo-comportamiento, determinar modelos
- **Control:** controlador, optimizador,
- **Clasificación:** economía, biología, procesamiento de imágenes, juegos



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

Algoritmos Genéticos

Algoritmos Genéticos (AGs)

METODO ESTOCASTICO DE BUSQUEDA INFORMADA DE SOLUCIONES CUASI-OPTIMAS. SE MANTIENE UNA POBLACION QUE REPRESENTA POSIBLES SOLUCIONES, LA CUAL ES SOMETIDA A CIERTAS TRANSFORMACIONES PARA OBTENER NUEVOS CANDIDATOS, Y A UN PROCESO DE SELECCIÓN SESGADO A FAVOR DE LAS MEJORES SOLUCIONES

Algoritmos Genéticos (AGs)

- Algoritmos de búsqueda que emulan la evolución biológica en el computador, siguiendo un proceso evolutivo inteligente sobre individuos,
- Fueron propuestos por John Holland a mediados de los años 70
- El objetivo final es encontrar la mejor solución posible, partiendo de soluciones escogidas aleatoriamente.

Algoritmos Genéticos (AGs)

- Dicha técnica es la más conocida.
- Razones:
 - Reúnen, de manera natural, todas las ideas fundamentales de la CE.
 - Poseen la mayor base teórica, la cual es sencilla y con mayores posibilidades de ampliación.
 - Requieren menor conocimiento específico del problema, lo que les da mayor versatilidad.

Características de los AGs

- Búsqueda Informada.
- Búsqueda Basada en Población.
- Operadores Aleatorios.

Generalidades de los AGs

- El procedimiento general consiste en mantener una población de potenciales soluciones a través de las generaciones.
- Los individuos de esa población son cadenas de caracteres.
- Se utilizan operadores evolutivos como mecanismos de búsqueda.
- El procedimiento continua hasta cumplirse algún criterio de convergencia o parada.
- Son el paradigma mas usado de la CE, y junto con las Redes Neuronales Artificiales, los mas usados de toda el área de Computación Inteligente.

Implementación de los AGs

- Seleccionar la Codificación: Si suponemos que un individuo Y posee m genes se requieren para la codificación L bits

$$L = \sum_{i=0}^m L_i \qquad L_i = \log_2 a_i \quad \text{gen } i \text{ tiene } a_i \text{ posibles valores (alelos).}$$

- Seleccionar la población inicial y su tamaño.
Lo mas variada posible.

Implementación de los AGs

- Definir criterio de parada

- *numero de iteraciones* (generaciones).
- *similitud de las estructuras*
- *similitud de contenido.*

- Funcion de Evaluación

De manera ideal, la función de evaluación coincide con el objetivo a maximizar o minimizar

- Manejar los individuos no factibles

- *penalización,*
- *reparación o corrección,*
- *decodificación*

Implementación de los AGs

- Seleccionar los operadores genéticos

La influencia de los operadores genéticos es fundamental en el funcionamiento de los AGs; por lo tanto, es importante hacer una buena selección de los mismos. Habitualmente, se utilizan dos operadores, el de cruce y el de mutación. No obstante, cuando se tiene conocimiento específico del problema es posible incorporar otros operadores genéticos, o modificar los operadores ya existentes.

Ejemplos de Aplicación

Búsqueda en un espacio de soluciones

- Supongamos que se desea maximizar la función: $F(x) = \text{Sen}(x\pi/256)$ en el intervalo $[0,255]$.
- *Codificación*. Como las soluciones varían entre 0 y 255, se pueden utilizar cadenas binarias de longitud 8. Por ejemplo, 00010100.

Ejemplos de Aplicación

- *Función de evaluación.* La función de evaluación es la misma función dada, y se utilizará como función de aptitud la diferencia entre el valor máximo (1) y el valor para la función dada: $g(x)=1-F(x)$.
- *Población inicial.*

Individuo	x	F(x)
10111101	189	0.733
11011000	216	0.471
01100011	99	0.937
11101100	236	0.243

Ejemplos de Aplicación

Problema de optimización combinatoria

- El *Agente Viajero*: Dadas N ciudades, el viajero de comercio debe visitar cada ciudad una vez teniendo en cuenta que el costo total del recorrido debe ser mínimo.

$G=(N, A)$

$N=\{1, \dots, n\}$: conjunto de n nodos (vértices)

$A=\{a_{ij}\}$: matriz de adyacencia.

$$d_{ij} = \begin{cases} \infty & \text{Si } a_{ij} = 0 \\ L_{ij} & \text{Si } a_{ij} = 1 \end{cases}$$

L_{ij} : distancia entre las ciudades i y j .

Ejemplos de Aplicación

- Suponiendo que las ciudades son numeradas desde 1 hasta n , una solución al problema puede expresarse a través de una matriz de estado E

$e_{ij} = \{ \begin{array}{l} 1 \text{ Si la ciudad } j \text{ fue la } i^{\text{ésima}} \text{ ciudad visitada} \\ 0 \text{ En otro caso} \end{array}$

- La matriz E permite definir un arreglo unidimensional V de dimensión n ;

$v_{ij} = i$ Si la ciudad i fue la $j^{\text{ésima}}$ ciudad visitada

Ejemplos de Aplicación

- Función objetivo

$$F1 = \sum_{i=1}^n \sum_{k=1}^n \sum_{j=1}^n L_{ik} e_{ij} e_{kj+1}$$

$$F2 = C \left(\left| \sum_{i=1}^n \sum_{j=1}^n e_{ij} - n \right| + \sum_{i=1}^n \left| \sum_{j=1}^n e_{ij} - 1 \right| + \sum_{j=1}^n \left| \sum_{i=1}^n e_{ij} - 1 \right| \right)$$

$$FC = F1 + F2$$

$C = n * \text{Max}(L_{ik})$ factor de penalización.

Ejemplos de Aplicación

- Codificación de los individuos:
 - Una solución viene dada como un recorrido que cumple con las restricciones del problema.
 - Se puede realizar por medio del arreglo unidimensional V de longitud n
- Función Objetivo: F1 o FC
- Operadores genéticos:
 - Si es F1 => Inversión.

LIMITACIONES

- FUNCION DE APTITUD SIEMPRE POSITIVA
- NUMERO DE GENERACIONES FINITO
- CODIFICACION BINARIA ES LA COMUN
- NO CONSIDERA RESTRICCIONES
- PROBLEMAS DE DIVERSIDAD: SUPERINDIVIDUOS, CONVERGENCIA PREMATURA, PRESION SELECTIVA, MECANISMOS DE ESCISION, CONTROLANDO EDADES, ETC.
- PROBLEMAS DE REPRESENTACION: COMPLETITUD, COHERENCIA, UNIFORMIDAD, ETC.

AVANCES

- FUNCIONES MULTIOBJETIVOS
=>conjunto de Paretos
- AG PARALELOS
- META-ALGORITMOS GENETICOS
=>APRENDIZAJE
- NICHOS Y ESPECIES

PROGRAMACIÓN GENÉTICA

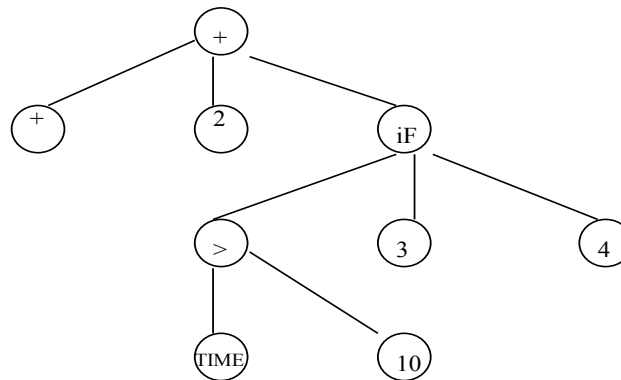
Consiste en utilizar la metodología de CE no para encontrar soluciones a problemas, sino para encontrar el mejor procedimiento para resolverlos.

Las estructuras que se someten a evolución codifican los algoritmos o programas que, al ser ejecutados, determinan soluciones.

PROGRAMACIÓN GENÉTICA

La representación de los individuos se hace a través de *árboles de análisis*.

En la practica, se usa el lenguaje LISP: Por ejemplo, la siguiente expresión: $(+ 1 2 (IF (> TIME 10) 3 4))$, su árbol es:



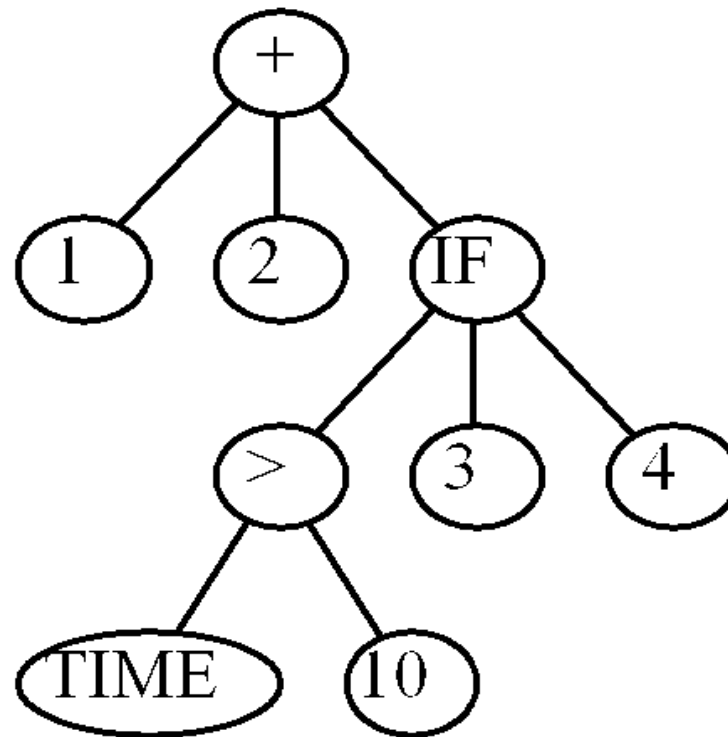
PROGRAMACIÓN GENÉTICA

```
int foo (int time)
{
    int temp1, temp2;
    if (time > 10)
        temp1 = 3;
    else
        temp1 = 4;
    temp2 = temp1 + 1 + 2;
    return (temp2);
}
```

Time	Output
0	6
1	6
2	6
3	6
4	6
5	6
6	6
7	6
8	6
9	6
10	6
11	7
12	7

PROGRAMACIÓN GENÉTICA

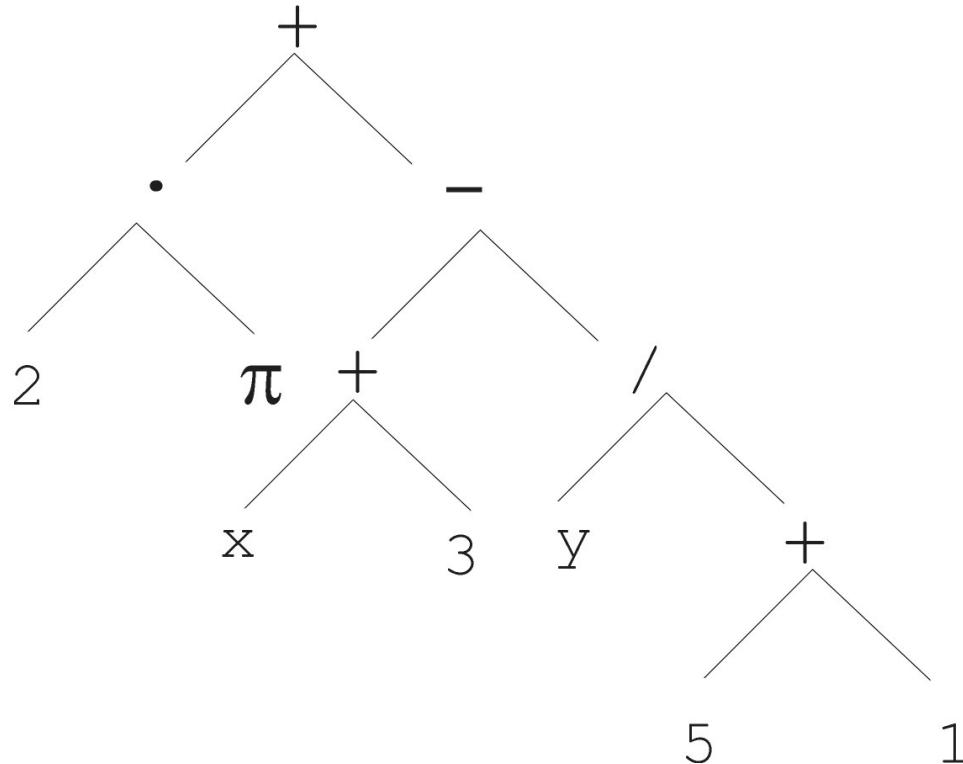
Representación de un programa



`(+ 1 2 (IF (> TIME 10) 3 4) )`

PROGRAMACIÓN GENÉTICA

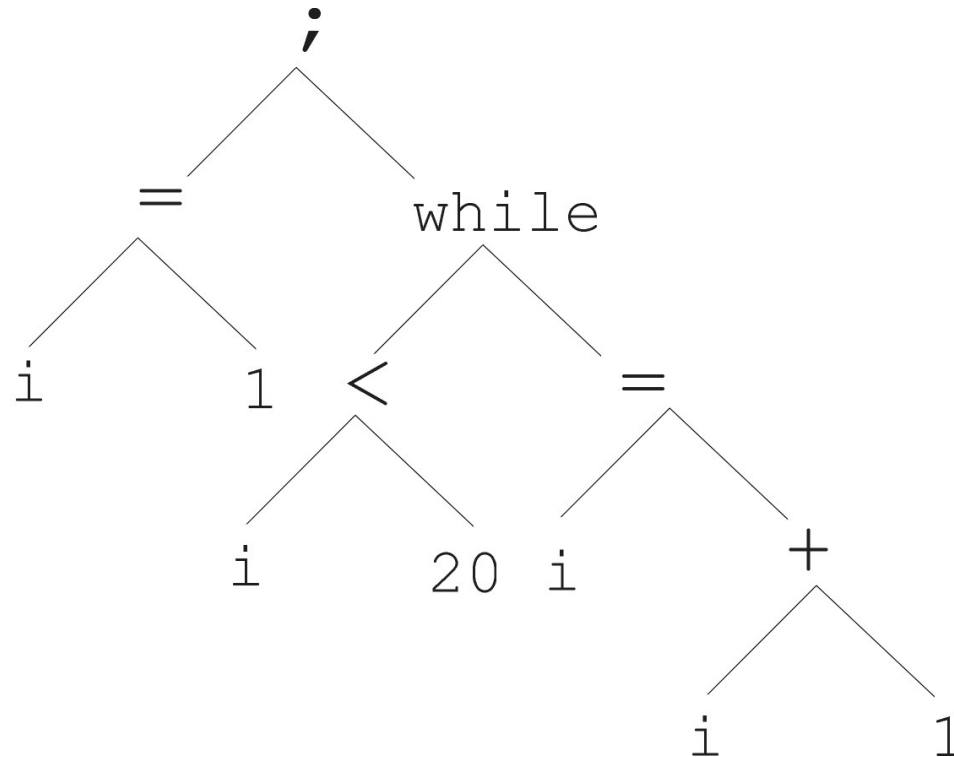
Representaciones de una fórmula



$$2 \cdot \pi + \left((x + 3) - \frac{y}{5 + 1} \right)$$

PROGRAMACIÓN GENÉTICA

Representaciones basadas en árboles



```
i = 1;  
while (i < 20)  
{  
    i = i + 1  
}
```

PROGRAMACIÓN GENÉTICA

La codificación utiliza un conjunto de *símbolos de funciones* y otro de *símbolos terminales* (átomos) adecuados para la solución del problema dado.

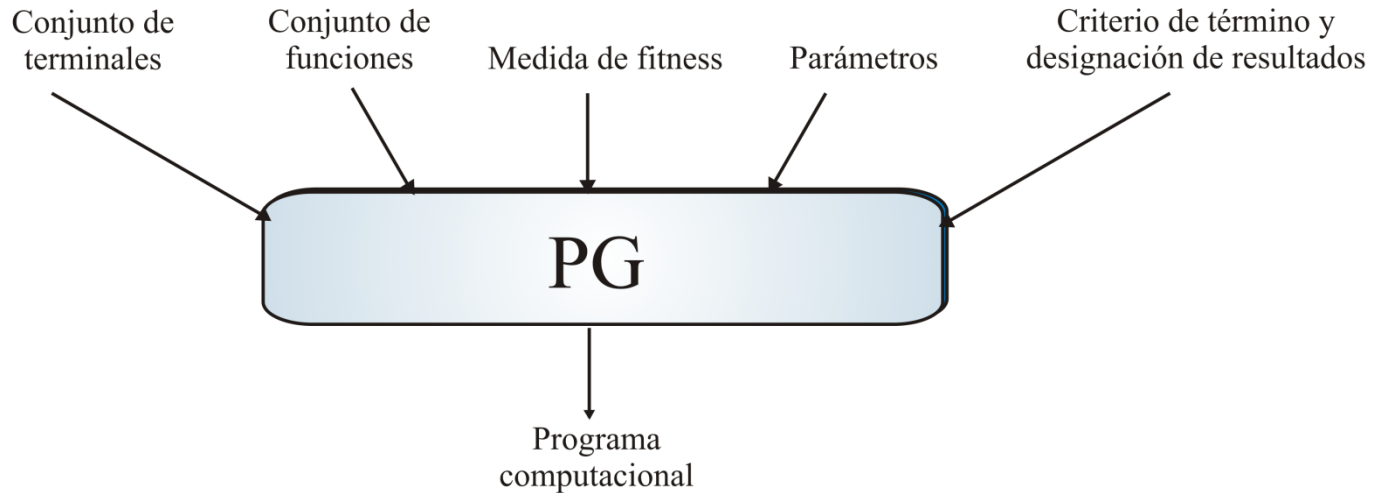
Las funciones pueden ser operaciones aritméticas, funciones matemáticas, funciones lógicas, o funciones específicas del dominio.

PROGRAMACIÓN GENÉTICA

ELEMENTOS:

- **CONJUNTO DE TERMINOS**
- **CONJUNTO DE FUNCIONES**
- **MEDIDA DE APTITUD**
- **PARAMETROS DE CONTROL**
- **CRITERIO DE CULMINACION**

PROGRAMACIÓN GENÉTICA



- Determinar el conjunto de terminales
- Determinar el conjunto de funciones
- Determinar la medición del fitness
- Determinar los parámetros para la ejecución
- Determinar el método para designar un resultado y un criterio para terminar una ejecución

Conjuntos de Funciones y Terminales

Los nodos en el árbol pueden ser:

- Conjunto de Funciones F
 - El conjunto de todos los posibles nodos internos en el árbol
 - Todas las funciones tienen una ariedad ≥ 1 .
- Conjunto de terminales T
 - El conjunto de todos los posibles nodos hoja en el árbol

El espacio de búsqueda del problema son todas las posibles combinaciones de funciones y terminales

Parámetros de control

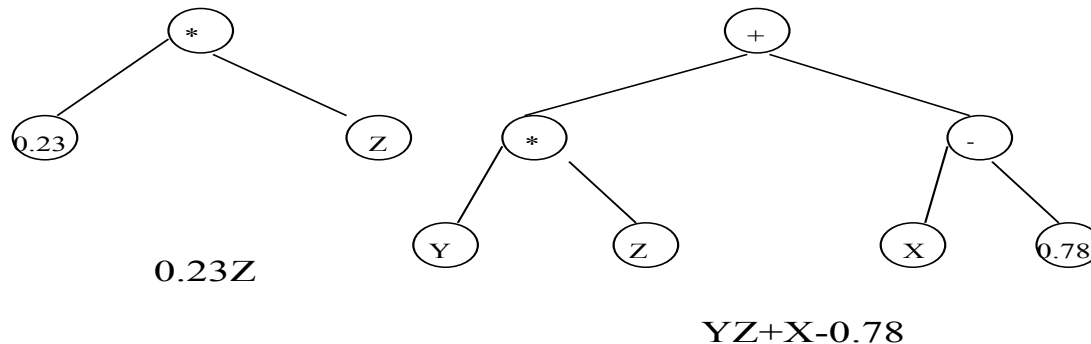
- Los parámetros de control más importantes son:
 - **G**: número máximo de generaciones a producir
 - **M**: tamaño de la población
 - **P_C**: probabilidad de cruzamiento
 - **P_M**: probabilidad de mutación
 - **D_{max}** tamaño máximo (medido por profundidad) de los individuos creados por las operaciones de evolución darwiniana
 - **D_{inicial}** tamaño (medido por profundidad) de los individuos creados en la población inicial

PROGRAMACIÓN GENÉTICA

EJEMPLO:

CONJ. TERMINOS: {X, Y, Z, REALES}

CONJ. FUNCIONES: {+, -, *, IF}



La función de Evaluación

- Función de Fitness “de Error”

$$f_p = \sum_{i=1}^n |p_i - o_i|$$

p : Algoritmo

p_i : la salida del programa p en el $i^{\text{ésimo}}$ caso de fitness

o_i : la salida del $i^{\text{ésimo}}$ ejemplo en el caso de fitness

f_p : el fitness de p.

n : número de casos de fitness

– Función de evaluación de error cuadrático

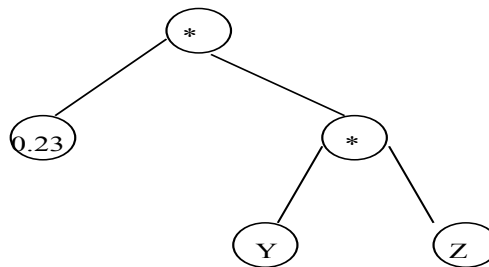
$$f_p = \sum_{i=1}^n (p_i - o_i)^2$$

PROGRAMACIÓN GENÉTICA

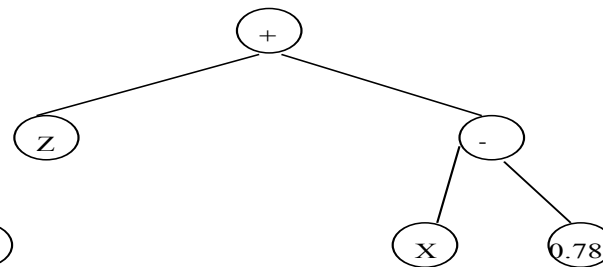
OPERADOR CRUCE

0.23Z YZ+X-0.78

=> 0.23YZ Z+X-0.78



0.23YZ

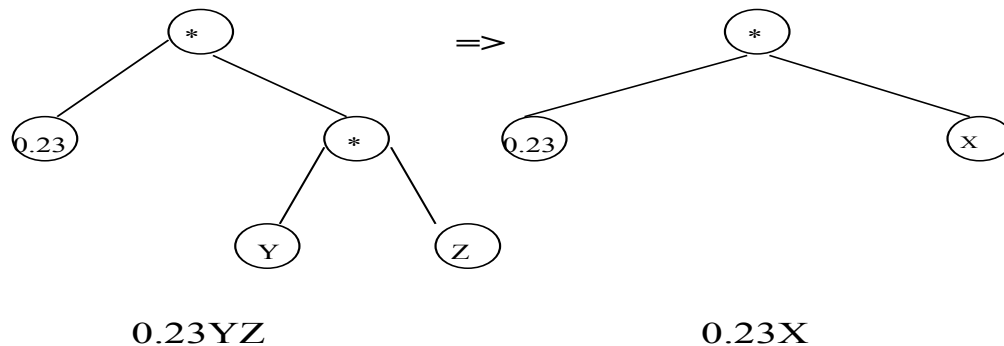


Z+X-0.78

PROGRAMACIÓN GENÉTICA

OPERADOR MUTACION

0.23YZ



PROGRAMACIÓN GENÉTICA

EJEMPLO: PROBLEMA DE REGRESION DE
FUNCION DESCONOCIDA

VAR. IND: X

VAR DEP: Y

<u>X</u>	<u>Y</u>
-1	0.178
-0.9	-0.169
...	
0	0.00
...	

PROGRAMACIÓN GENÉTICA

PROBLEMA: ENCONTRAR $Y=F(X)$

TERMINOS: X

FUNCIONES: +, -, *, SIN COS, EXP, LOG

FUNC. ADAPTAT.: ERROR

GENER 0: $X + \text{LOG}(2X) + X$ fa: 6.05

$\text{COS}(\text{COS}(+(-(*XX))))$ fa: 23,67

GENER 34: $X^4 + X^3 + X^2 + X$ fa: 0

PROGRAMACIÓN GENÉTICA

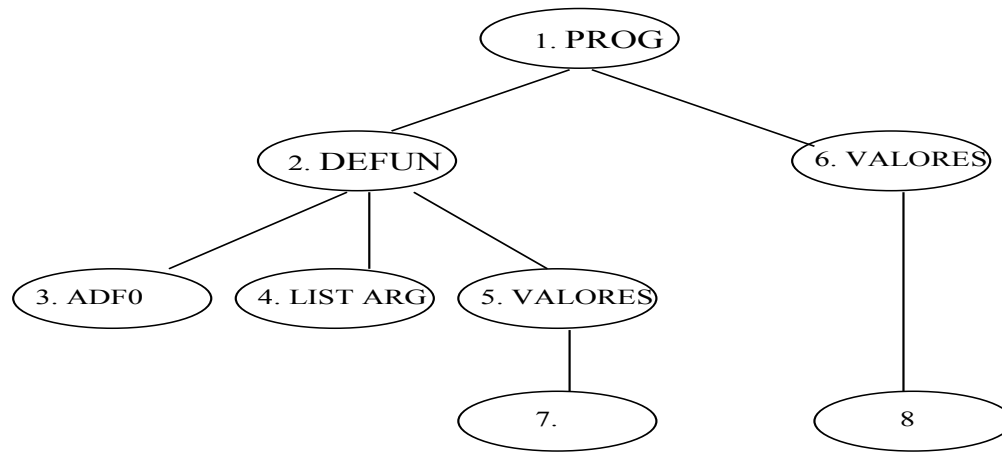
ADF (Automatically Defined Functions)

- DESCUBRIMIENTO AUTOMATICO DE UNIDADES FUNCIONALES
- INSPIRACION: HUMANOS ESCRIBEN FUNCIONES/RUTINAS QUE AGRUPAN PORCIONES DE INSTRUCCIONES QUE PUEDEN SER LLAMADAS VARIAS VECES DESDE PROGRAMAS
 - DESCOMPOSICION DE PROBLEMAS
 - RESOLUCION DE SUBPROBLEMAS
 - RECUPERACION DE LAS SOLUCIONES PARCIALES

PROGRAMACIÓN GENÉTICA

ADF (Automatically Defined Functions)

- DEFINICION DE UN ADF



1. RAIZ

2. INICIO RAMA DEF. FUNCION

3. NOMBRE ADF

4. LISTA ARGUMENTOS

5. VALORES A REGRESAR

6. VARIAB. QUE GUARDAN RES.

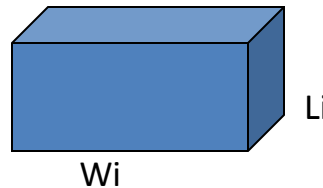
7. CUERPO ADF

8. CUERPO PROGRAMA

PROGRAMACIÓN GENÉTICA

EJEMPLO ADF

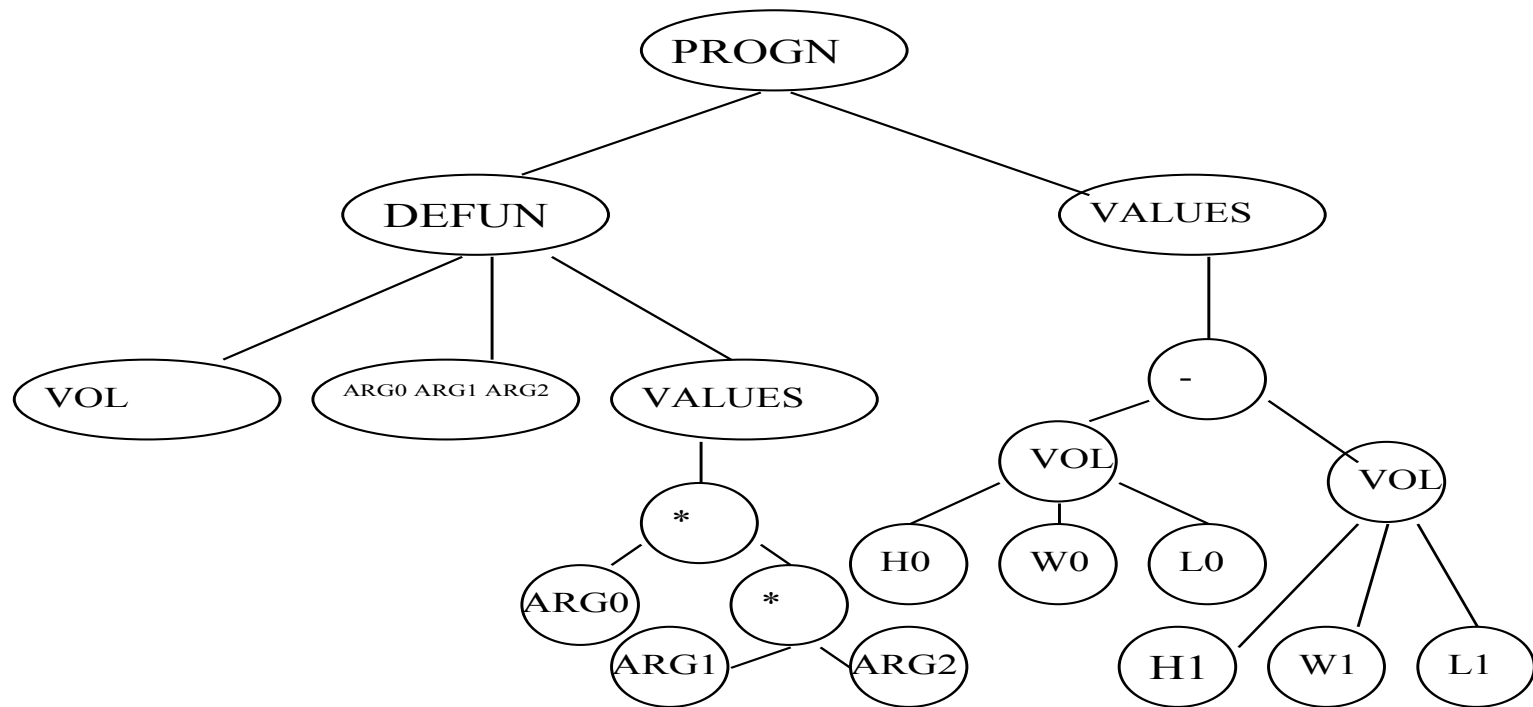
- PROGRAMA CALCULA DIF ENTRE 2 CUBOS



ENFOQUE CLASICO: $(-(*(* W0 L0)) H0) (* (* W1 L1) H1))$

ENFOQUE CON ADF:
(PROG (DEF (VOL ARG0 AR1 ARG2)
VALUES (*ARG0 (* ARG1 ARG2))))
(VALUES (- VOL L0 W0 H0)
(VOL L1 W1 H1))))

PROGRAMACIÓN GENÉTICA



PROGRAMACIÓN GENÉTICA

ELEMENTOS DE UN ADF

- NUMEROS DE ADFs
- SUS ARGUMENTOS
- LAS REFERENCIAS JERARQUICAS
- OPERADORES
 - CREACION: ADF O INVOCACION
 - DUPLICACION: ADF O INVOCACION
 - ELIMINACION: ADF O INVOCACION

PROGRAMACIÓN GENÉTICA

AREAS DE APLICACIÓN

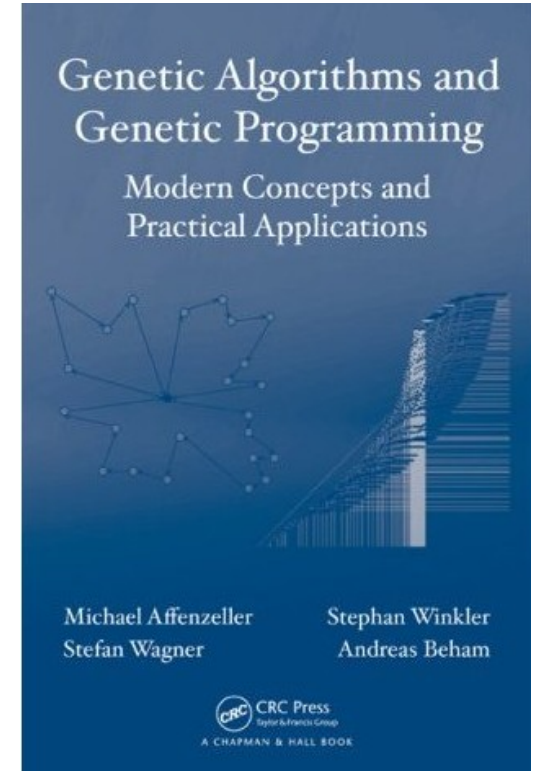
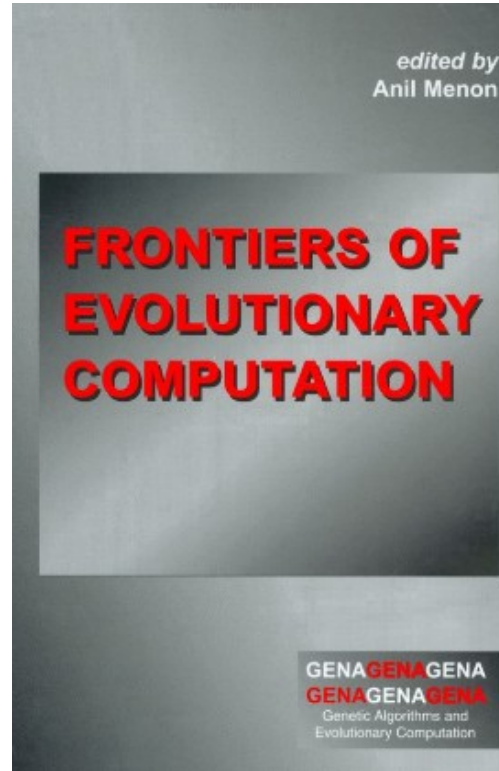
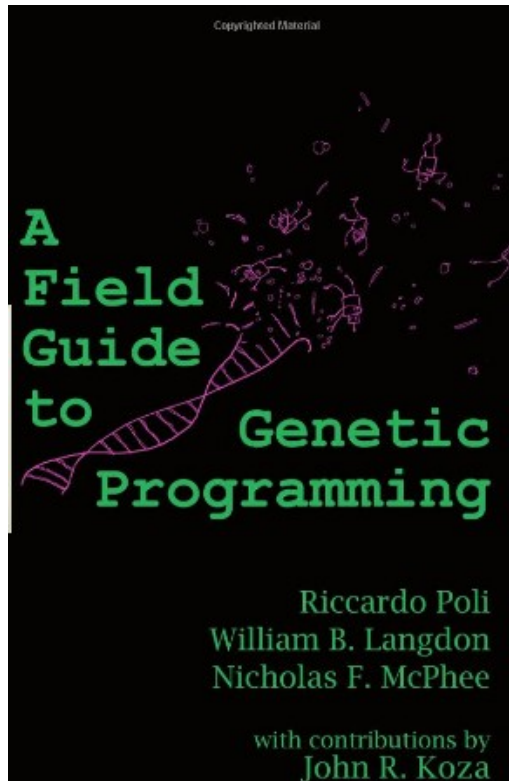
- CONTROL: IDENTIFICACION, ETC.
- MINERIA DE DATOS
- SISTEMAS MULTIAGENTES
- HARDWARE EVOLUTIVO

www.aic.nrl.navy.mil/galist

Kernel GPC++

Es una biblioteca de clases escritas en el lenguaje de programación orientado a objetos C++, que permite utilizar la técnica de Programación Genética con un esfuerzo mínimo, dado que el *kernel* proporciona el conjunto de estructuras, clases y algoritmos necesarios para utilizar la técnica.

Algunas Referencias





Aspectos básicos de los sistemas lógicos difusos

Jose Aguilar Castro
CEMISID

Justificación

Aquellos casos en los cuales **la complejidad del problema y su carácter dinámico, o las incertidumbres alrededor del mismo**, hacen difícil el establecimiento detallado de especificaciones. Parecen ser más apropiados adoptar enfoques que usen el **aprendizaje por medio de la experiencia o el razonamiento heurístico** para la toma de decisiones que permitan producir “soluciones confiables”

Lógica Difusa

- En la lógica clásica una proposición sólo admite dos valores: puede ser verdadera o falsa.

Lógica binaria

- Existen otras lógicas que admiten un tercer valor:

posiblemente

- La lógica difusa (o borrosa) es una de ellas, que se caracteriza por **querer cuantificar esta incertidumbre**

Lógica difusa

- Los conjuntos difusos fueron introducidos como una forma matemática de representar la “**vaguedad**” de la vida cotidiana.
- La idea fundamental es la de desarrollar un marco de trabajo para el **tratamiento de la imprecisión**.
- Nació alrededor del año 1965 con los trabajos de Zadeh en los Estados Unidos, en los cuales aplicó la **lógica multivaluada** a la teoría de conjuntos.

Lógica Difusa

- Zadeh introdujo el uso de una función, que expresaba el **grado de pertenencia** (o membresía) de los elementos a un conjunto dado, con valores en el rango entre 0 y 1.
- Estos últimos años ha tenido un gran interés en la comunidad científica a raíz de los avances en **lógica difusa, los algoritmos difusos, el control difuso, el razonamiento difuso**, etc.

Areas de aplicación

- Los sistemas basados en lógica difusa **imitan la forma en que toman decisiones los humanos**, con la ventaja de ser mucho más rápidos.
- Son generalmente robustos y tolerantes a imprecisiones y ruidos en los datos de entrada.
- En la lógica difusa, **se usan modelos matemáticos para mapear nociones subjetivas**, como *caliente/tibio/frío*, a valores concretos que puedan ser manipuladas por los computadores.
- La lógica difusa se utiliza cuando la complejidad del proceso en cuestión es muy alta y no existen modelos matemáticos precisos: procesos altamente no lineales, cuando se envuelven definiciones/conocimiento no estrictamente definidos (o subjetivos).

Áreas de aplicación de la L.D.

Productos de Consumo	Automóviles y generación eléctrica	Procesos Industriales	Robótica y Manufactura
Cámaras y filmadoras (Cannon, Minolta, Ricoh, Sanyo) Lavadoras (Gldstar, LG, Whirpool) Refrigeradoras Aspiradoras	Control de vehículos (Nissan) Control de transmisión (Mitsubishi, Mazda, Honda)	Refinación Destilación Plantas petroquímicas Incineración Producción de cemento	Robots (mitsubishi) Reconocimiento de patrones y Visión por ordenador

LA BORROSIDAD

Por borrosidad entendemos el hecho de que una proposición pueda ser parcialmente verdadera y parcialmente falsa de forma simultánea.

- Replanteamiento radical de conceptos clásicos de verdad y falsedad, por el concepto de vaguedad o borrosidad. La verdad y/o falsedad no son más que casos extremos.
- Una persona no será simplemente alta o baja, sino que participará de ambas características parcialmente, mientras que en la zona intermedia de ambas alturas existirá una gradualidad por la que va dejando de ser alta.
- El concepto de borrosidad está enraizado en la mayor parte de nuestros modos de pensar y hablar.

Lógica Difusa

DIFERENCIAS ENTRE PROBABILIDAD Y BORROSO:

PROBABILIDAD:

- NÚMERO ENTRE 0 Y 1 PARA EXPRESAR LA POSIBLE OCURRENCIA DE UN EVENTO (NO HA OCURRIDO).
- SUPONE TENER UN MODELO DEL MUNDO.
- ES BASADA EN FRECUENCIAS HISTORICAS

BORROSO:

- GRADO DE POSIBILIDAD QUE UN PREDICADO SEA CIERTO.
- NO SE CONOCE MODELO DEL MUNDO.
- USA DESCRIPCIONES Y DESCRIBE AL MUNDO

Lógica Difusa

LÓGICA MULTIVALUADA EN LA TEORÍA DE CONJUNTOS

=> GRADOS DE PERTENENCIA

=> CLASES CON LÍMITES MAL DEFINIDOS

⇒ GRADUALIDAD EN LOS CAMBIOS DE ESTADOS

⇒ ETIQUETAS LINGÜÍSTICAS

Lógica Difusa

- **Conjuntos difusos:** son una generalización de los conjuntos ordinarios pero agregándoles un grado de pertenencia a cada elemento.
- **Grado de pertenencia** varia entre 0 y 1.
 - 0 significa que ese elemento no pertenece a un conjunto dado,
 - 1 significa que el elemento pertenece 100% a ese conjunto.

Conjuntos difusos

Un conjunto difuso posee como lógica subyacente una **lógica multivaluada**, y permite la **descripción de conceptos** en los cuales la **transición** entre poseer una propiedad y no poseerla, no son claros.

Definición de conjunto ordinario

Sea X un universo y sea S un subconjunto de X .

La función característica asociada con S es un mapa:

$$\mu_s : X \rightarrow \{0,1\}$$

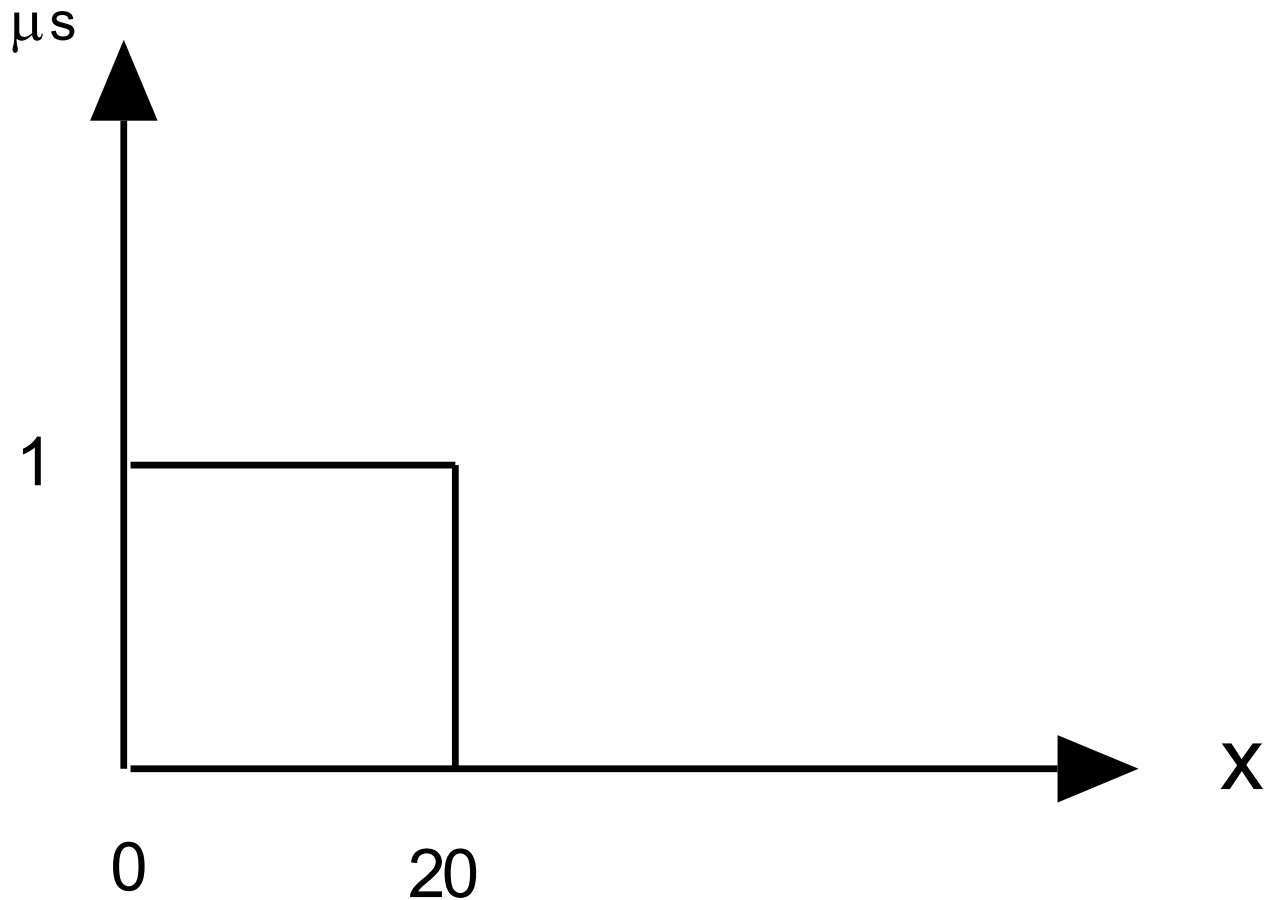
tal que para cualquier elemento x del universo X ,

si x es un miembro de S

$$\mu_s(x) = 1 \text{ si } x \text{ no es un miembro de } S$$

$$\mu_s(x) = 0$$

Ejemplo de conjunto ordinario



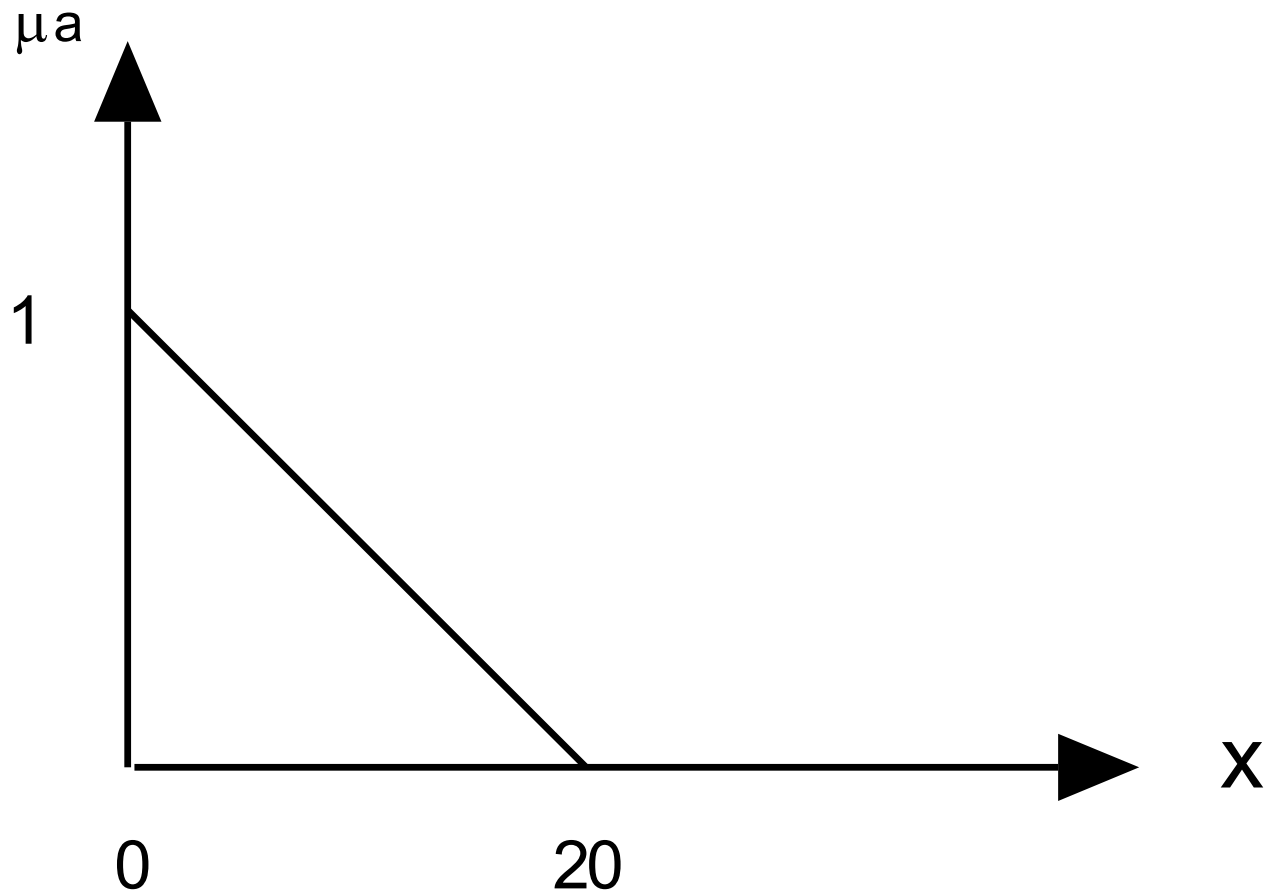
Definición de conjunto difuso

Sea X es un conjunto que representa un universo. Un subconjunto difuso A del universo X esta asociado a una función característica.

$$\mu_A : X \rightarrow [0,1]$$

$\mu_A(x)$ indica el grado con el cual el elemento x del universo X es miembro del conjunto A

Ejemplo de conjunto ordinario



Lógica Difusa

Representación de los conjunto difuso

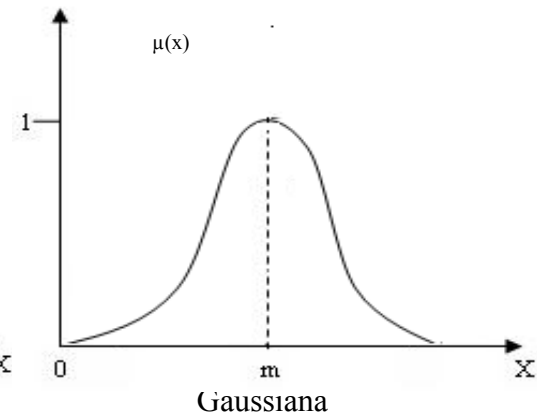
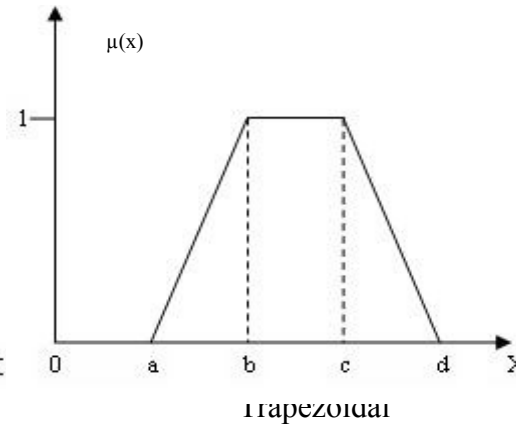
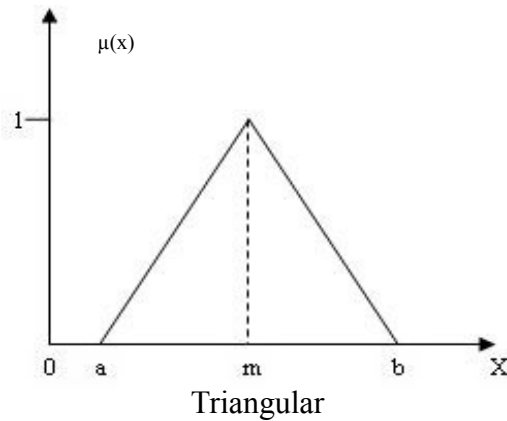
$A = \{x / \mu_A(x) \text{ para } x \in X\}$, donde x es un elemento, $\mu_A(x)$ define la función de pertenencia y X es el Universo.

- **Función de pertenencia para un conjunto:** define el grado de pertenencia de cada elemento a ese conjunto.

Función de membrecía

- En la teoría de conjuntos difusos, la función característica de un conjunto es llamada *función de membrecía o pertenencia*,
- Indica el grado con el que un elemento x_i pertenece a un conjunto X .

Ejemplos de funciones de membrecía



$$\mu_A(x) = \exp\left(\frac{-(x - \mu_A(x))^2}{2\sigma_A^2}\right)$$

Donde σ permite cambiar la forma

Ejemplo de Conjunto Difusa con funciones de membrecía

- U es universo recurso **EDAD** [0,120]

$$U = \{x/x \in [0, 120]\}$$

- Conjunto difusos **joven** y **viejo**

$$\text{Joven} = \{(x/1, 0 \leq x \leq 40) \cup \{x/1 + (x-40) / 40\}, x > 40)\}$$

$$\text{Viejo} = \{(x/0, 0 \leq x \leq 40) \cup \{x/1 + (x-40) / 40\}, x > 40)\}$$

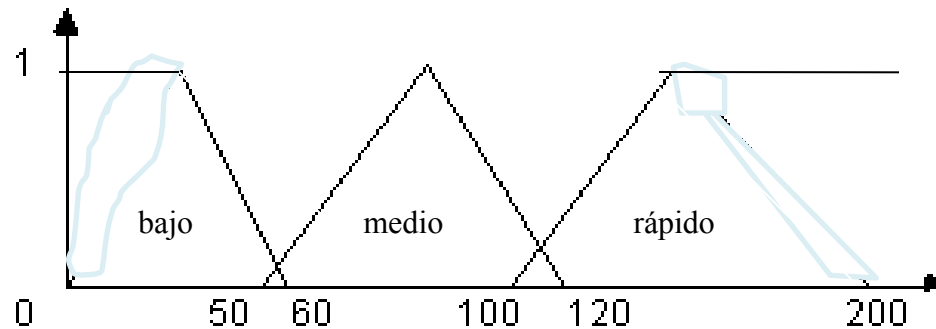
Nomenclatura de conjuntos difusos

Asuma que $X = \{x_1, x_2, x_3, x_4\}$; entonces $A = \left\{ \frac{0.7}{x_1}, \frac{0.3}{x_2}, \frac{1}{x_3}, \frac{0}{x_4} \right\}$ provee una representación de un subconjunto difuso de X .

En esta representación del subconjunto difuso A , un término de la forma a_1 / x_1 debe ser entendido como indicando que el elemento x_1 posee grado de membresía a_1 en el subconjunto difuso A .

EJEMPLO DE CONJUNTOS DIFUSOS

- 3 CONJUNTOS DIFUSOS:
- RÁPIDO [100,200] , LENTO[0,60] , MEDIA[50,120]



- PARA UN CARRO QUE VA A UNA VELOCIDAD DE 55 KM/HORA:

$$\mu_{\text{BAJO}}(55) = 0.25 \quad \mu_{\text{MEDIO}}(55) = 0.25 \quad \mu_{\text{RAPIDO}}(55) = 0$$

$$\text{Bajo} = \{0/0, 0.5/15, 1/30, 0.5/45, 0/60\}$$

Variables Lingüísticas

- Si una **variable** puede tomar **valores** de “**palabras**” en un lenguaje natural (por ejemplo, pequeño, rápido, etc.), entonces esa variable define una **variable lingüística**.
- Estas palabras generalmente son **etiquetas** de conjuntos difusos.
- Una variable lingüística toma **valores lingüísticos**(por ejemplo, para velocidad: bajo, rápido, etc.); pero cuando es instanciada toma números según **función de pertenencia**).

Variables Lingüísticas

- Variable lingüística = $(x, A(x), U, G, M)$

Donde

- X es el nombre de la variable
- $A(x)$ es el conjunto de términos lingüísticos de x
- U es el universo donde se define cada valor de x
- G es la regla sintáctica para generar las sentencias correctas en A
- M es la regla semántica que asocia a cada valor x su significado

Variables Lingüísticas

- Ejemplo:
 - X es la variable lingüística *edad*
 - $A(edad) = \{\text{niño, joven, adulto, anciano}\}$
 - Universo de discurso $U [0, 120]$
 - G es la regla sintáctica para generar conjunto A
 - M es la regla semántica que asocia a cada valor x su significado

Lógica Difusa

- **Operaciones difusas**, para los conjuntos difusos A y B en el mismo universo U, tenemos (solo ultimo caso no):

– **Unión:**

$$\mu_{A \cup B} = \mu_A(x) \vee \mu_B(x) = \{x / \max(\mu_A(x), \mu_B(x)) \quad \forall x \in U\}$$

– **Intersección:**

$$\mu_{A \cap B} = \mu_A(x) \wedge \mu_B(x) = \{x / \min(\mu_A(x), \mu_B(x)) \quad \forall x \in U\}$$

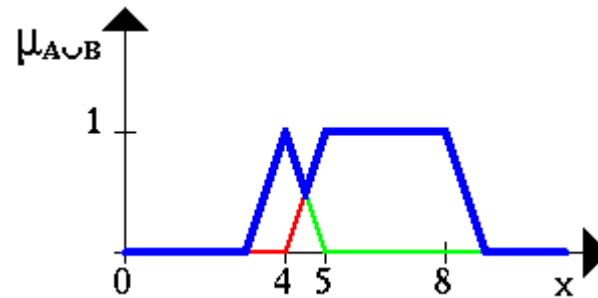
– **Complemento de** $\mu_A(x) = \{x / (1 - \mu_A(x)) \quad \forall x \in U\}$

– **Producto cartesiano:**

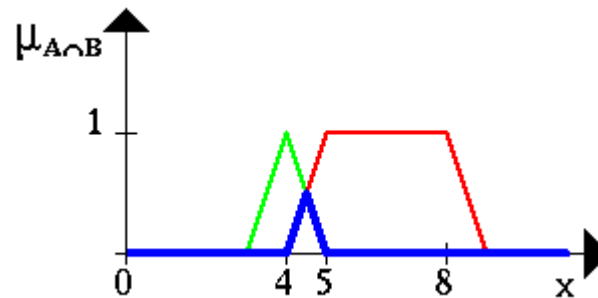
$$\mu_{A \times B}(x, y) = \{(x, y) / \min(\mu_A(x), \mu_B(y)) \quad \forall x \in U, \forall y \in V\}$$

Lógica Difusa

– Unión:

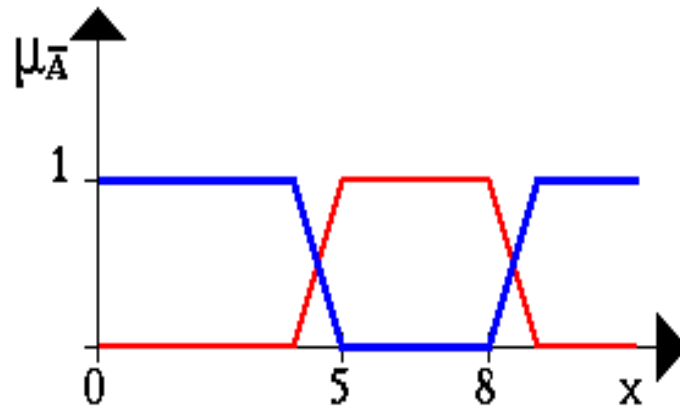


– Intersección:



Lógica Difusa

– Negación:



Operaciones sobre conjuntos difusos

Asuma que A y B son subconjuntos difusos de X. Su unión es un subconjunto difuso C de X, denotado por $C=A \cup B$, tal que para cada $x \in X$

$$C(x)=\text{Max}[A(x),B(x)]=A(x) \vee B(x)$$

Ejemplo: Sea $X=\{a,b,c,d,e\}$; asuma que

$$A = \{1/a, 0.7/b, 0.3/c, 0/d, 0.9/e\} \text{ y } B = \{0.2/a, 0.9/b, 0.4/c, 1/d, 0.4/e\}$$

$$\text{Entonces } C = \{1/a, 0.9/b, 0.4/c, 1/d, 0.9/e\}$$

Operaciones sobre conjuntos difusos

Asuma que A y B son subconjuntos difusos de X , la intersección de A y B , denotado por $D=A \cap B$, es tal que para cada $x \in X$,

$$D(x) = \text{Min}[A(x), B(x)] = A(x) \wedge B(x)$$

Ejemplo: Considerando A y B del ejemplo anterior, entonces $D = \{0.2/a, 0.7/b, 0.3/c, 0/d, 0.4/e\}$

Operaciones sobre conjuntos difusos

Teorema: Asuma que A y B son subconjuntos difusos de X . Si $C=A\cup B$ y $D=A\cap B$ entonces

$$D \subset C$$

$$A \subset C \quad \text{y} \quad B \subset C$$

$$D \subset A \quad \text{y} \quad D \subset B$$

Operaciones sobre conjuntos difusos

Asuma que A y B son subconjuntos difusos de X . El complemento relativo de B respecto de A , denotado por $E=A-B$, es definido como el subconjunto difuso E de X donde para cada $x \in X$.

$$E(x) = \text{Max}[0, A(x) - B(x)]$$

Ejemplo: Asuma que

$$A = \{1/a, 0.3/b, 0.7/c, 0.6/d, 1/f\} \text{ y}$$
$$B = \{1/a, 1/b, 0.3/c, 0/d, 0.5/f\}.$$

Entonces

$$E = A - B = \{0/a, 0/b, 0.4/c, 0.6/d, 0.5/f\}.$$

Operaciones sobre conjuntos difusos

Conmutatividad

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

Asociatividad

$$A \cup (B \cup C) = (A \cup B) \cup C = A \cup B \cup C$$

$$A \cap (B \cap C) = (A \cap B) \cap C = A \cap B \cap C$$

Distributividad

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Operaciones sobre conjuntos difusos

- IDEMPOTENCIA:

$$A \vee A = A$$

$$A \wedge A = A$$

- BORNES UNIVERSALES:

$$A \vee 0 = A$$

$$A \vee 1 = 1$$

$$A \wedge 0 = 0$$

$$A \wedge 1 = A$$

Operaciones sobre conjuntos difusos

Propiedades que cumplen el complemento y el complemento relativo de subconjuntos difusos A y B de X .

Doble negación

$$\overline{\overline{A}} = A$$

Leyes de Morgan

$$\overline{(A \cup B)} = \overline{A} \cap \overline{B}$$

$$\overline{(A \cap B)} = \overline{A} \cup \overline{B}$$

$$\overline{\emptyset} = X ; \overline{X} = \emptyset$$

Operaciones sobre conjuntos difusos

- Las funciones **min** y **max** son generalizaciones de las operaciones homónimas de los conjunto clásicos.
- En general:
 - La unión e intersección borrosa requieren funciones de forma tal que al aumentar uno de los conjuntos también aumente su unión e intersección
 - Dichas funciones deben ser conmutativas, distributivas y continuas

Las funciones que verifican tales propiedades son las que pertenecen a las clases conormas triangulares (conorma T) y normas triangulares (norma T)

Relaciones difusas

Sean U y V dos universos de discurso. Una relación difusa R es un conjunto difuso en el espacio producto $U \times V$. Esto es, R posee una función de membrecía $\mu_R(u,v)$, donde $u \in U$ y $v \in V$

Ejemplo: Si $X = \{a,b,c\}$ y $Y = \{1,2\}$ entonces

$$A = \left\{ \frac{0.1}{(a,1)}, \frac{0.6}{(a,2)}, \frac{0.9}{(b,1)}, \frac{1}{(b,2)}, \frac{0}{(c,1)}, \frac{0.2}{(c,2)} \right\}.$$

es una relación difusa sobre el espacio X y Y .

Relaciones difusas

Si A_1, A_2, A_k, A_n son subconjuntos difuso de x_1, x_2, x_k, x_n respectivamente, entonces sus productos cruzados $A_1 \times A_2 \times A_k \times A_n$ es un subconjunto difuso de $x_1 \times x_2 \times x_k \times x_n$, denotado por $T = A_1 \times A_2 \times A_k \times A_n$, donde

$$T(x_1, x_2, x_k, x_n) = \min_i [A_i(x_i)], \quad x_i \in X_i \quad i=1, 2, \dots, n.$$

Ejemplo: Asuma que $A = \{1/a, 0.6/b, 0.3/c\}$ y $B = \{1/1, 0.5/2, 0/3\}$ son subconjuntos difusos sobre los espacios $X = \{a, b, c\}$ y $Y = \{1, 2, 3\}$ respectivamente, entonces

$$A \times B = \left\{ 1/(a,1), 0.5/(a,2), 0/(a,3), 0.6/(b,1), 0.5/(b,2), 0/(b,3), 0.3/(c,1), 0.3/(c,2), 0/(c,3) \right\}$$

Relaciones difusas

Asuma que A es una relación difusa sobre $X \times Y$. La proyección de A sobre X es un subconjunto difuso A° de X , que denotamos por $A^\circ = \text{Proy}_x A$ y que definimos por $A^\circ(x) = \text{Max}_y [A(x, y)]$

Ejemplo: Asuma $X = \{a, b, c\}$ y $Y = \{1, 2, 3\}$ y sea $A = \{1/(a,1), 0.6/(a,2), 0.4/(a,3), 0.5/(b,1), 0.8/(b,2), 0.2/(b,3), 0.3/(c,1), 0.1/(c,2), 0.3/(c,3)\}$

Entonces $\text{Proy}_x A = \{1/a, 0.8/b, 0.3/c\}$ y $\text{Proy}_y A = \{1/1, 0.8/2, 0.3/3\}$

Medidas Borrosas

- Borrosidad: distancia de un conjunto difuso A a uno discreto C, tal que C contiene los valores de x para $\mu_A(x) > 0$
- Distancia entre dos conjuntos difusos A y C
 - Hamming $f(A) = \sum |\mu_A(x) - \mu_C(x)|$
 - Euclidea $f(A) = (\sum ((\mu_A(x) - \mu_C(x))^2))^{1/2}$
 - ...
- Similitud es igual a distancia
- Entropía Borrosa: cuanto aporta conjunto A a la descripción de la variable x

$$f(A) = \sum \{\mu_A(x) \log \mu_A(x) + [1 - \mu_A(x)] \log [1 - \mu_A(x)]\}$$

Operaciones sobre conjuntos difusos

- **Reglas de modificación:** adverbio (bastante, muy, mas o menos, etc.) que se utiliza para especificar una propiedad mas concreta
- Son operaciones sobre la función de pertenencia, ejemplos:
 - **Negación o “no”:** $\text{NEG}(\mu(x)) = 1 - \mu(x)$
 - **Concentración o “muy”:** $\text{CON}(\mu(x)) = \mu(x)$
 - **Dilatación o difusión o “algo”, o “mas o menos”² o “casi”:**

$$\text{DIL}(\mu(x)) = \mu(x)$$
 - **Intensificación o “bastante”:**

$$\text{INT}(\mu(x)) = \begin{cases} 2\mu(x) & \text{si } 0 \leq \mu(x) \leq 0.5 \\ 1 - 2(1 - \mu(x)) & \text{si } \mu(x) > 0.5 \end{cases}$$

Operaciones sobre conjuntos difusos

- Altura de un conjunto difuso: valor mas grande de su conjunto de pertenencia $\sup_{x \in X} \mu_A(x)$
- Conjunto difuso normalizado: $\text{Altura}(A)=1$
- Soporte de un conjunto difuso ($\text{soporte}(A)$): elementos de X que pertenecen a A con grado mayor a 0
- Núcleo de un conjunto difuso: elementos de X que pertenecen a A con grado mayor = 1
- Cardinalidad de un conjunto difuso $\text{card}(A) = \sum_{x \in X} \mu_A(x)$
- α -corte: valores de X con grado menores a α
- Inclusión difusa

$$S(A,B) = \sum \max(0, \mu_A(x) - \mu_B(x)) / \text{card}(A)$$

Aspectos básicos de Lógica Difusa

- Un conjunto difuso A de X es llamado normal si existe al menos un elemento

$$x \in X / A(x) = 1$$

Un conjunto difuso que no es normal es llamado sub-normal.

Aspectos básicos de Lógica Difusa

Ejemplo:

Asuma que $X=\{a,b,c,d,e\}$ y sean A y B subconjuntos difusos de X.

$$A = \{1/a, 0.3/b, 0.2/c, 0.8/d, 0/e\}$$

$$B = \{0.6/a, 0.9/b, 0.1/c, 0.3/d, 0.2/e\}$$

- A es un subconjunto normal, mientras que B es subnormal
- $\text{Altura}(A)=1$ y $\text{Altura}(B)=0.9$
- $\text{Soporte}(A)=\{a,b,c,d\}$ y $\text{Soporte}(B)=\{a,b,c,d,e\}$
- $\text{Núcleo}(A)=\{a\}$ y $\text{núcleo}(B)=\emptyset$

Aspectos básicos de Lógica Difusa

Asuma que A y B son subconjuntos difusos de X .

Decimos que A es un subconjunto de B ,

denotado por $A \subset B$

si $B(x) \geq A(x)$

para cada $x \in X$

Ejemplo: Asuma $A = \{0.1/a, 0.6/b, 1/c\}$ y $B = \{0.3/a, 0.9/b, 1/c\}$,
entonces $A \subset B$.

Aspectos básicos de Lógica Difusa

- Un subconjunto difuso nulo de X es definido como un subconjunto difuso $\emptyset$ de X tal que $\emptyset(x)=0$ para cada $x \in X$.
- La potencia del subconjunto A , denotada por $\text{potencia}(A)$, es definida como

$$\text{Potencia}(A) = \sum_{i=1}^n A(x_i)$$

Operaciones sobre conjuntos difusos

Asuma que A es un subconjunto difuso de X y α es cualquier numero no negativo; entonces A^α representara al conjunto difuso B de X tal que

$$B(x) = (A(x))^\alpha$$

Ejemplo: Asuma que $A = \{1/a, 0.6/b, 0.3/c, 0/d, 0.5/e\}$ entonces

$$A^2 = \{1/a, 0.36/b, 0.09/c, 0/d, 0.25/e\}$$

Puede verse fácilmente que si $\alpha > 1$ entonces $A^\alpha \subset A$ y si $\alpha < 1$ entonces $A^\alpha \supset A$.

Operaciones sobre conjuntos difusos

Asuma que A es un subconjunto difuso de X ; el conjunto de nivel α de A , denotado por A_α , es un subconjunto ordinario de X consistente de todos los elementos en X para los cuales $A(x) \geq \alpha$. Esto es

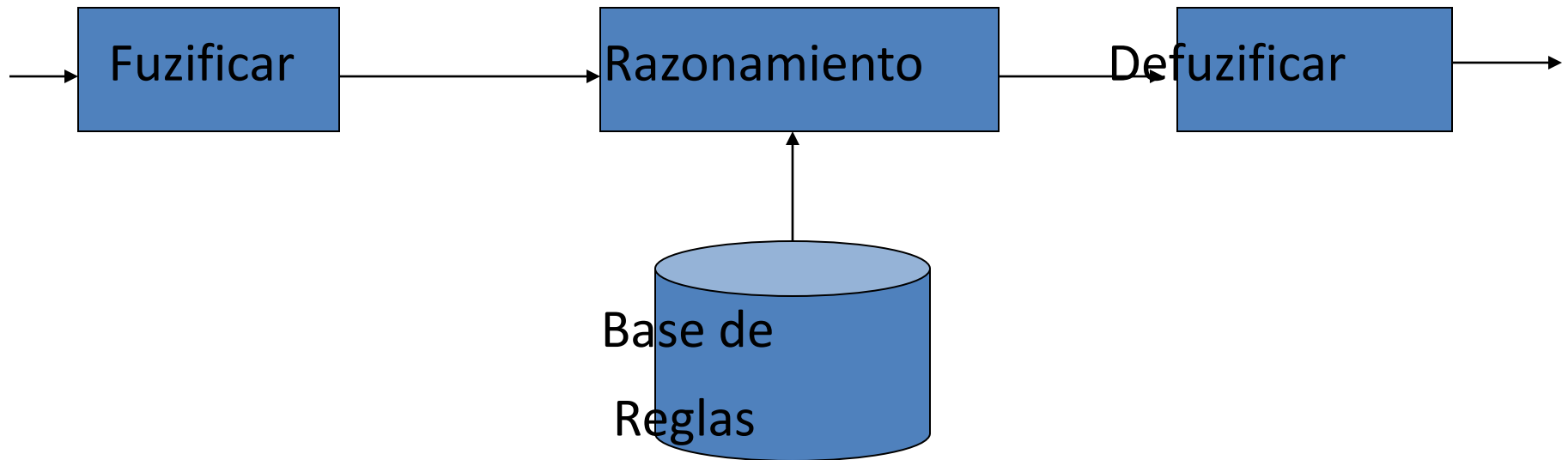
$$A_\alpha = \{x / A(x) \geq \alpha, x \in X\}$$

Ejemplo: Asuma que $X = \{a, b, c, d, e\}$ y
 $A = \{0.3/a, 1/b, 0.5/c, 0.9/d, 1/e\}$, en este caso

$$A_\alpha = \{a, b, c, d, e\} \quad \text{para} \quad \alpha \geq 0.3$$

$$A_\alpha = \{b, d, e\} \quad \text{para} \quad \alpha \geq 0.9$$

Sistemas Difusos



Base de Reglas

- La lógica difusa puede ser representada por implicaciones difusas del tipo

$A \Rightarrow B$ (condición/acción)

si A entonces B, donde A y B son **proposiciones difusas**.

- La inferencia borrosa es basada en **implicaciones borrosas** y las reglas de inferencia derivadas de estas

*$\Rightarrow$ proceso deductivo llamado **modus ponens**.*

Por ejemplo, si *Ana es Joven* entonces *personalidad es inmadura*.

Proposiciones Difusas

- Proposición X es A ,

donde X es una variable que toma su valor de un dominio U y A es un conjunto difuso que pertenece a ese dominio

La talla de Juan es Grande

- Imprecisión no difusa

Juan Mide entre 1.80m y 1.90m

- Imprecisión difusa

Juan mide casi 1.82m

Elementos de un Sistema de Razonamiento aproximado

- Los elementos básicos de un sistema de razonamiento aproximado son la colección de **variables simples** $V_1, \dots, V_n$ y una colección de conjuntos $X_1, \dots, X_n$, donde cada X_i es el *conjunto base* en el universo de discurso de la variable V_i .
- Dicho conjunto X_i contiene los valores posibles que puede tomar la variable V_i .

Elementos de un Sistema de Razonamiento aproximado

- Una *variable conjunta* es una colección de una o más variables simples. Por ejemplo, V_a, V_b y V_c son variables conjuntas definidas por $V_a = (V_1, V_2)$, $V_b = (V_1, V_4, V_5)$ y $V_c = V_1$.
- Una *proposición* es una sentencia de la forma $V \text{ es } M$, donde V es una variable conjunta, M es un conjunto difuso sobre el espacio del producto cartesiano de los conjuntos base de las variables simples que conforman a la variable conjunta.

Elementos de un Sistema de Razonamiento aproximado

- El producto cartesiano “ x ” de los conjuntos base de las variables V_i está definido sobre el universo de V . Así, M es una relación difusa sobre X . Una proposición de la forma dada es llamada una *sentencia canónica*.
- Por ejemplo: Si V_1 representa la presión de un tanque y V_2 el nivel, entonces la variable conjunta $V=(V_1,V_2)$ tiene como conjunto base los pares dados por la presión y el nivel.

Elementos de un Sistema de Razonamiento aproximado

- Si las variables en la operación conjunción son las mismas, la operación se reduce a la operación *intersección* entre conjuntos difusos.

$$(Va \text{ es } M) \cap (Va \text{ es } N) = Va \text{ es } M \cap N$$

- Si Va y Vb no tienen variables comunes, la operación conjunta resulta en el clásico producto cartesiano. Esto es:

$$(Va \text{ es } M) \cap (Vb \text{ es } N) = (Va, Vb) \text{ es } M \times N$$

La manipulación del conocimiento a través de proposiciones difusas en un sistema de razonamiento aproximado, está basada fundamentalmente en las operaciones de *implicación* y *proyección*.

Ejemplo Producto Cartesiano

Por ejemplo, sean V_1 , V_2 y V_3 variables simples sobre los conjuntos base X , Y y Z , respectivamente. Si definimos dos variables conjuntas $V_a=(V_1, V_2)$ y $V_b=(V_2, V_3)$, y un par de proposiciones “ V_a es M ” y “ V_b es N ”, donde M y N son conjuntos difusos dados por:

$$M = \left\{ \frac{0.7}{(x_1, y_1)}, \frac{0.9}{(x_1, y_2)}, \frac{1}{(x_2, y_1)}, \frac{0.2}{(x_2, y_2)} \right\}$$

$$N = \left\{ \frac{0.8}{(y_1, z_1)}, \frac{1}{(y_1, z_2)}, \frac{0.4}{(y_2, z_1)}, \frac{1}{(y_2, z_2)} \right\}$$

La conjunción de estas proposiciones es la proposición “ V es P ”, donde P es el conjunto difuso dado por:

$$P = \left\{ \frac{0.7}{(x_1, y_1, z_1)}, \frac{0.7}{(x_1, y_1, z_2)}, \frac{0.4}{(x_1, y_2, z_1)}, \frac{0.9}{(x_1, y_2, z_2)}, \right. \\ \left. \frac{0.8}{(x_2, y_1, z_1)}, \frac{1}{(x_2, y_1, z_2)}, \frac{0.2}{(x_2, y_2, z_1)}, \frac{0.2}{(x_2, y_2, z_2)} \right\}$$

Implicaciones difusas

Una implicación difusa $A \rightarrow B$ puede ser entendida como una regla difusa del tipo “Si – Entonces”.

Ejemplo: Si x es A entonces y es V ; donde $x \in U$, $y \in V$ son variables lingüísticas.

Las expresiones (a) - (i) corresponden a interpretaciones de las reglas “Si – Entonces” basadas en criterios intuitivos o en generalizaciones de la lógica clásica.

Modelos Difusos

Basado en Reglas SI-ENTONCES con proposiciones difusas y razonamiento aproximado.

- SISO SI A ENTONCES B
- MISO SI U1 Y U2 Y U3 ... ENTONCES B
- MIMO SI U1 Y U2 Y U3 ... ENTONCES V1; V2; V3

Razonamiento Aproximado Básico (CADA REGLA TIENE SU PESO)

- :SI (X ES A_i) ENTONCES (Y SERÁ C_j)

$$\mu_{C_j}(Y) = \text{PESO DE LA REGLA} = \mu_{A_i}(X)$$

- SI (X_1 ES A_i) Y (X_2 ES B_j) ENTONCES (Y SERÁ C_j)

$$\mu_{C_j}(Y) = \text{PESO DE LA REGLA} = \min(\mu_{A_i}(X_1), \mu_{B_j}(X_2))$$

- SI (X_1 ES A_i) O (X_2 ES B_j) ENTONCES (Y SERÁ C_m)

$$\mu_{C_m}(Y) = \text{PESO DE LA REGLA} = \max(\mu_{A_i}(X_1), \mu_{B_j}(X_2))$$

- R1: SI ... ENTONCES (Y1 SERÁ C) Y (Y2 SERA D)

R2: SI ... ENTONCES (Y1 SERÁ C) Y (Y2 SERA E)

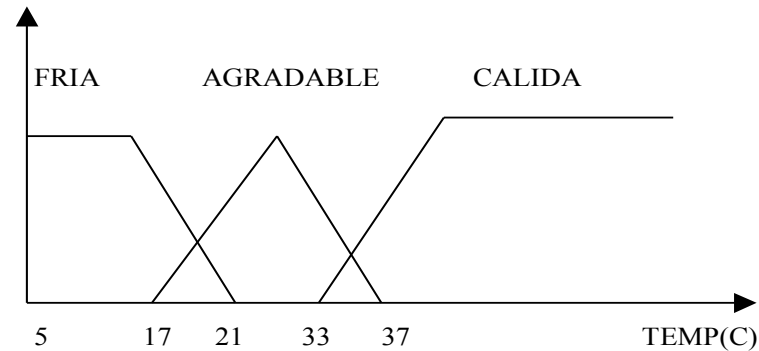
$$\mu_C(R1 \cup R2) = \max(\text{PESO REGLA R1}, \text{PESO REGLA R2})$$

$$\mu_D(Y1) = \text{PESO REGLA R1}$$

$$\mu_E(Y2) = \text{PESO REGLA R2}$$

Lógica Difusa

- Ejemplo 1:

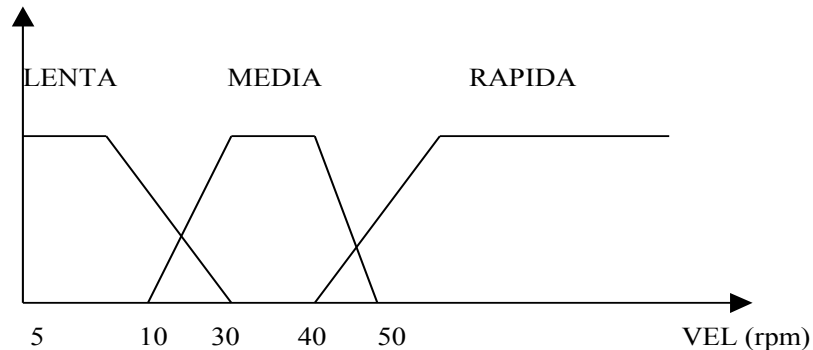


$TEMP(18) = \{0.5/FRIA, 0.2/AGRADABLE, 0/CALIENTE\}$

BASE DE REGLAS: SI TEMP ES FRIA ENTONCES VEL SERA LENTA

SI TEMP ES AGRADABLE ENTONCES VEL SERA MEDIA

SI TEMP ES CALIDA ENTONCES VEL SERA MAXIMA



Algoritmo de inferencia para modelo SISO

SI (X ES Bi) ENTONCES (Y SERÁ Di)

1. Para cada Regla

- Grado de disparo de la regla i (nivel de disparo τ_i de la i-ésima regla)

$$\tau_i = \mu_{B_i}(x)$$

- Conjunto difuso F_i dado como salida por la regla i (implicación difusa)

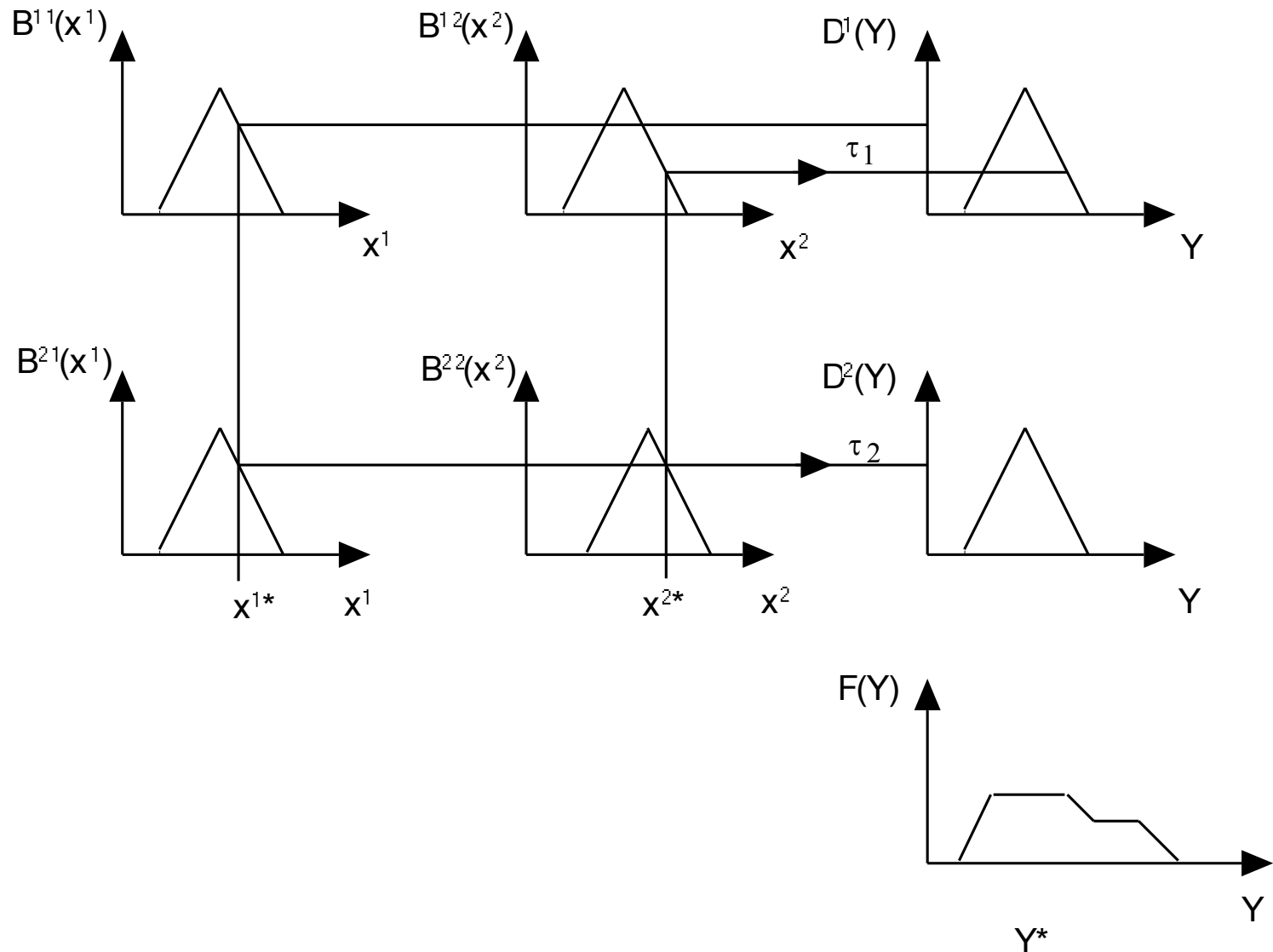
$$F_i(y) = \tau_i \wedge \mu_{D_i}(y) \quad \Rightarrow \quad \mu_{R_i}(x, y) = \mu_{B_i}(x) \wedge \mu_{D_i}(y)$$

2. Agregación de las salidas de cada regla $F(y)$

$$F_i \quad F(y) = \bigvee_i F_i(y) \quad R = \bigcup_i R_i$$

$$\Rightarrow \quad \mu_R(x, y) = \bigvee_i \mu_{R_i}(x, y) = \bigvee_i (\mu_{B_i}(x) \wedge \mu_{D_i}(y))$$

Ilustración gráfica del algoritmo de razonamiento de un SD tipo Mamdani



Lógica Difusa (Modelo de Mamdani)

- EJEMPLO 2:

R1: SI e ES NEG Y ∇e ES NEG ENTONCES ∇u ES NEG

R2: SI e ES NEG Y ∇e ES CERO ENTONCES ∇u ES NEG

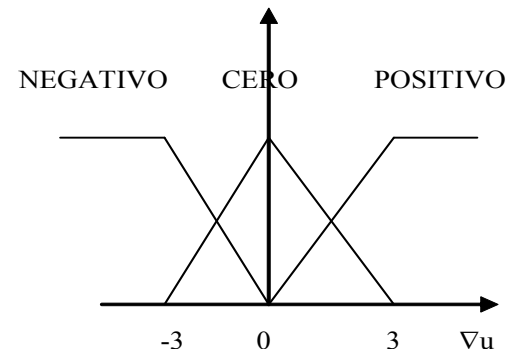
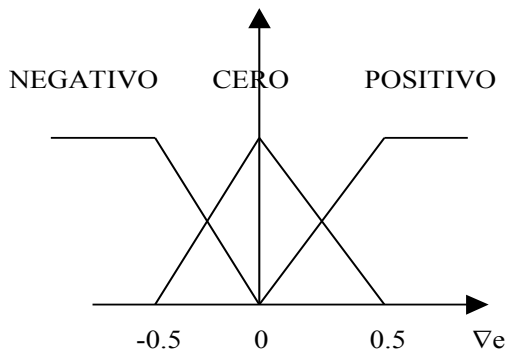
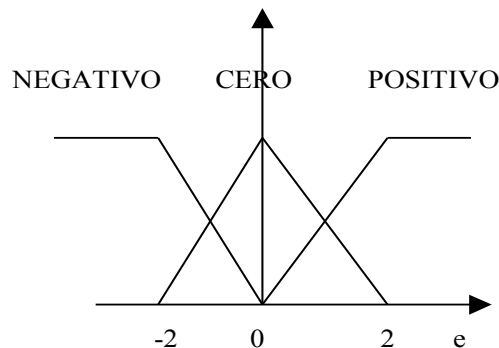
R3: SI e ES NEG Y ∇e ES POSIT ENTONCES ∇u ES CERO

R4: SI e ES CERO Y ∇e ES NEG ENTONCES ∇u ES NEG

R5: SI e ES CERO Y ∇e ES CERO ENTONCES ∇u ES CERO

R6: SI e ES CERO Y ∇e ES POSIT ENTONCES ∇u ES POSIT

...



Lógica Difusa (Modelo de Mamdani)

- EJEMPLO 2:

SUPONER DE ENTRADA $e = -2.1$ Y $\nabla e = 0.5$

=> R3 UNICA QUE SE ACTIVA

ADEMAS, EL UNIVERSO DE DISCURSO DISCRETO DE

∇e ES $\{-2, -1, -0.5, -0.25, 0, 0.25, 0.5, 1, 2\}$

e ES $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$

∇u ES $\{-6, -4.5, -3, -1.5, 0, 1.5, 3, 4.5, 6\}$

$$\begin{aligned} \Rightarrow \quad \mu_N(\nabla u) &= [1, 1, 1, 0.5, 0, 0, 0, 0, 0] \\ \mu_C(\nabla u) &= [0, 0, 0, 0.5, 1, 0.5, 0, 0, 0] \\ \mu_P(\nabla u) &= [0, 0, 0, 0, 0, 0.5, 1, 1, 1] \end{aligned}$$

GRADO MEMBRESIA DE LOS CONJUNTOS DIFUSOS DE SALIDAS

$$\mu_{F1}(\nabla u) = [0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\mu_{F2}(\nabla u) = [0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\mu_{F3}(\nabla u) = [0, 0, 0, 0.5, 1, 0.5, 0, 0, 0] \quad \mu_{F4}(\nabla u) = [0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\mu_{F5}(\nabla u) = [0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\mu_{F6}(\nabla u) = [0, 0, 0, 0, 0, 0, 0, 0, 0]$$

DEFUSIFICACIÓN

CONSISTE EN DETERMINAR VALOR NUMÉRICO
DESDE UNA SALIDA BORROSA.

- FUNCION DE PERTENENCIA DEL CONJUNTO BORROSO DE SALIDA

$$\begin{aligned}\mu_X(Y) &= \text{PESO REGLA } R_i * \mu_{X_i}(Y) + \text{PESO REGLA } R_j * \mu_{X_j}(Y) + \dots \\ &= \text{MIN}(\text{PESOREGLA } R_i, \mu_{X_i}(Y)) + \text{MIN}(\text{PESOREGLA } R_j, \mu_{X_j}(Y))\end{aligned}$$

Defusificación sencilla

- Dado un valor puntual $x^* \in X$, el mecanismo de difusificación consiste en crear un conjunto difuso A , cuya función de pertenencia es aquella donde para cualquier $x \in X$ su valor es cero excepto en el valor de x^* , donde toma el valor de 1. Esto es:

$$\mu_A(x) = \begin{cases} 0 & \text{Si } x \neq x^* \\ 1 & \text{Si } x = x^* \end{cases}$$

Nótese que este conjunto difuso así considerado no es mas que un conjunto ordinario con un único elemento dado por x^*

TÉCNICAS CLÁSICAS

DEFUZIFICACIÓN

- **CENTROIDE:** TRATA DE CALCULAR EL CENTRO DE MASA

$$Y = \sum_K Y_K \mu(Y_K) / \sum_K \mu(Y_K)$$

- **MÁXIMO:** MAYOR VALOR DE SALIDA Y SE LE CALCULA SU CENTROIDE

$$Y = \text{MAX} (Y_1 \rightarrow \mu(Y_1), Y_2 \rightarrow \mu(Y_2), \dots)$$

Método del Centro del Area (MCA)

- El MCA define el valor defusificado de un conjunto difuso F como su centroide difuso. Esto es

$$y^* = \frac{\int_{\mathcal{R}} yF(y)dy}{\int_{\mathcal{R}} F(y)dy}$$

Método del Centro del Area (MCA)

- El cálculo del valor desdifusificado con el MCA es simplificado si consideramos un universo finito Y. Así, $F(y)$ será una función de membresía discreta:

$$y^* = \frac{\sum_{j=1}^n F(y_j) y_j}{\sum_{j=1}^n F(y_j)}$$

Método de Media de Máximos (MMM)

- El MMM determina el valor desdifusificado como una media de todos los valores del universo que poseen grado de membresía máximo. Esto es

$$y^* = \frac{1}{q} \sum_{j \in J^*} y_j$$

Lógica Difusa

- Ejemplo 1:

$$Y = [\text{CENTROIDE}_{\text{LENTA}} \mu_{\text{LENTA}}(Y) + \text{CENTROIDE}_{\text{MEDIA}} \mu_{\text{MEDIA}}(Y)] /$$
$$[\mu_{\text{LENTA}}(Y) + \mu_{\text{MEDIA}}(Y)] =$$
$$(10 * 0.5 + 35 * 0.2) / (0.5 + 0.2) = 17,14 \text{rpm}$$

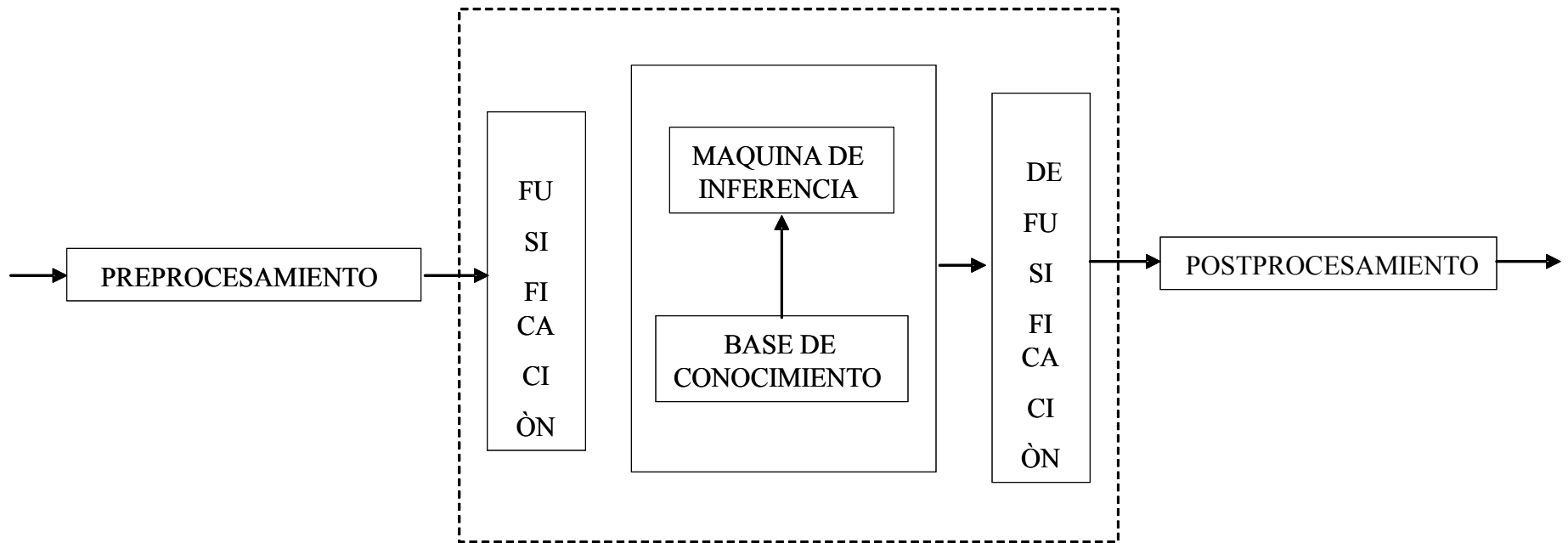
- EJEMPLO 2:

$$\nabla_u = [0.5(-1.5) + 1(0) + 0.5(1.5)] / (0.5 + 1 + 0.5) = 0$$

Definición de Fusificador

El fusificador realiza una transformación de un punto de un conjunto ordinario $X=(x_1, x_2, \dots, x_n) \in U$ a un conjunto difuso $A' \in U$.

componentes básicos en el diseño de un sistema lógico difuso



Lógica Difusa

Pasos a seguir para aplicar la lógica difusa en un problema:

- Identificar las variables y sus posibles rangos de valores.
- Determinar las funciones de pertenencia de esos valores a expresiones descriptivas.
- Determinar las reglas que rigen el comportamiento del sistema.
- Seleccionar algún método para darle un valor preciso a los resultados descriptivos.

Fuentes de información para sistemas lógicos difusos

- En la mayoría de los sistemas ingenieriles, existen dos fuentes importantes de información:
 - La proveniente de sensores
 - La proveniente de la experticia humana.
- La incorporación sistemática y eficiente de información lingüística facilita el diseño de aplicaciones con miras a incorporar características como adaptación, aprendizaje y autonomía.

Incorporación de información en Sistemas Lógicos Difusos

- La incorporación de información numérica y lingüística en el diseño de aplicaciones de lógica difusa en procesos, puede ser realizada siguiendo alguno de los métodos siguientes:
 - Usando la experiencia de expertos y conocimientos de Ingeniería.
 - Basado en las acciones que realizan los operadores.
 - Basado en auto-organización o aprendizaje.

A continuación se presenta una
Metodología para el diseño de
aplicaciones usando sistemas
lógicos difusos

Etapas 1: Análisis y descripción del problema.

Fase 1.1.- Descripción General del Problema:

- 1.1.1.- Familiarización con el proceso sobre el cual se desea realizar el Sistema Lógico Difuso.
- 1.1.2.- Familiarización con los ambientes computacionales donde se encuentran los datos a ser utilizados.
- 1.1.3.- Definición detallada del problema que motiva el desarrollo del Sistema Lógico Difuso.

Etapa 1: Análisis y descripción del problema.

- Fase 1.2.- Análisis de Factibilidad para el desarrollo del Sistema Lógico Difuso.
- Fase 1.3.- Análisis de datos
- Fase 1.4.- Elección de la fuente de conocimiento

Etapa 2: Especificación de requerimientos

- Fase 2.1.- Estimación del perfil de los usuarios finales del Sistema Lógico Difuso.
- Fase 2.2.- Determinación de los requerimientos de información.
- Fase 2.3.- Determinación de los requerimientos funcionales

Etapa 2: Especificación de requerimientos

- Fase 2.4.- Determinación de los requerimientos de entrada de datos:
 - 2.4.1.- Selección de las posibles variables de entrada al Sistema Lógico Difuso.
 - 2.4.2.- Identificación de las fuentes de datos.
 - 2.4.3.- Especificación de los procesos de adquisición de datos.
 - 2.4.4.- Especificación de los procesos de generación de parámetros.
 - 2.4.5.- Caracterización de la interoperabilidad entre las bases de datos que se requieren en la implantación.

Etapa 2: Especificación de requerimientos

- Fase 2.5.- Definición de los requerimientos de hardware y software para la implantación del Sistema Lógico Difuso:
 - 2.5.1.- Especificación de la plataforma de hardware que se utilizará para el desarrollo y operación del Sistema Lógico Difuso.
 - 2.5.2.- Determinación, análisis y selección de las herramientas de software disponibles en el mercado para el desarrollo de Sistema Lógico Difuso.
- Fase 2.6.- Verificación de los requerimientos con el usuario.

Etapa 3: Análisis de costos, tiempo y recursos.

- Fase 3.1.- Elaboración del plan de actividades de desarrollo e implantación.
- Fase 3.2.- Estimación del tiempo requerido para el desarrollo del Sistema Lógico Difuso.
- Fase 3.3.- Estimación de los recursos computacionales (hardware-software) requeridos para el desarrollo del Sistema Lógico Difuso.
- Fase 3.4.- Estimación de los costos de desarrollo.

Etapa 4: Diseño de Sistema Lógico Difuso

- Fase 4.1.- Pre-procesamiento, puede involucrar:
 - Cuantificación en conexión con el muestreo y redondeo.
 - Normalización o escalamiento a un rango estandarizado.
 - Filtraje de las señales para la remoción de ruidos.
 - Pre-mediación para la obtención de tendencias de corto y largo plazo.
 - Combinación de varias mediciones para obtener indicadores claves.
 - Diferenciación e integración o sus equivalentes discretos

Etapas 4: Diseño de Sistema Lógico Difuso.

- Fase 4.2.- Fusificación: En esta fase cada dato de entrada es convertido a grados de membrecía asignado a una o varias funciones de membrecía.
- Fase 4.3.- Base de Reglas: Se modela lingüísticamente el sistema o proceso. Las reglas pueden utilizar varias variables, tanto en sus antecedentes como en sus consecuentes.

Etapas 4: Diseño de Sistema Lógico Difuso.

- Fase 4.4.- Motor de Inferencia: En esta fase se construye el mecanismo de toma de decisiones. El motor de inferencia permite establecer los valores de membresía en las variables del antecedente y del consecuente de cada regla, a objeto de establecer la salida que proporciona la base de reglas ante las entradas dadas.

Etapas 4: Diseño de Sistema Lógico Difuso.

- Fase 4.5.- Defusificación: Esta fase involucra la transformación de los valores difusos, que proporciona el motor de inferencia, en valores ordinarios.
- Fase 4.6.- Post-procesamiento: En esta fase, la salida defusificada definida en un universo estándar debe ser escalada en términos de unidades ingenieriles.

Etapas 5: Diseño asociado a la herramienta computacional.

- Fase 5.1.- Diseño preliminar de la arquitectura del Sistema computacional.
- Fase 5.2.- Selección de la herramienta computacional de acuerdo a los requerimientos surgidos en la etapa de diseño del sistema lógico difuso.
- Fase 5.3.- Diseño preliminar de los procesos de adquisición y almacenamiento de datos

Etapas 5: Diseño asociado a la herramienta computacional.

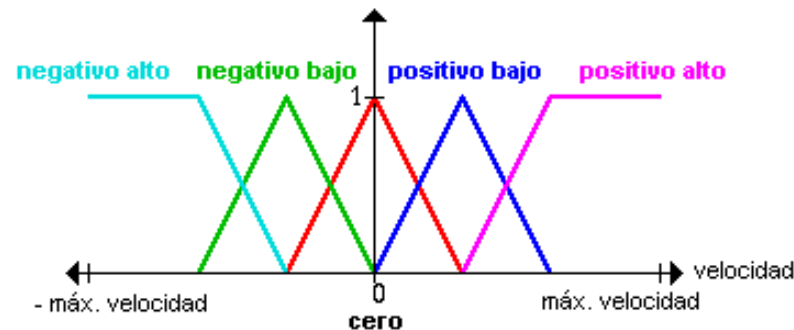
- Fase 5.4.- Diseño preliminar de los procesos de interconexión.
 - 5.4.1.- Integración Interna.
 - 5.4.2.- Integración Externa.
 - 5.4.2.- Selección de software auxiliar.
- Fase 5.5.- Verificación del diseño preliminar del Sistema computacional

Etapas 6: Desarrollo e Implantación del Sistema Lógico Difuso

- Fase 6.1.- Construcción del prototipo.
- Fase 6.2.- Validación del prototipo.
- Fase 6.3.- Construcción del modelo operacional
- Fase 6.4.- Prueba y depuración
- Fase 6.5.- Mantenimiento y actualización.

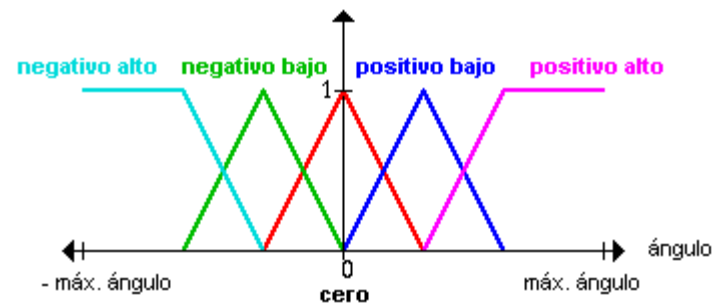
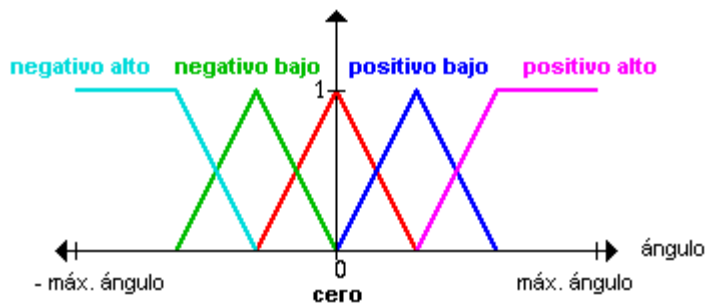
Ejemplo

- **El péndulo invertido:** El problema está en equilibrar una pértiga sobre una plataforma móvil que puede moverse en dos únicas direcciones, a la izquierda o a la derecha.
- Se define (subjetivamente) cual es la velocidad del anden: alta, baja, etc.



Ejemplo

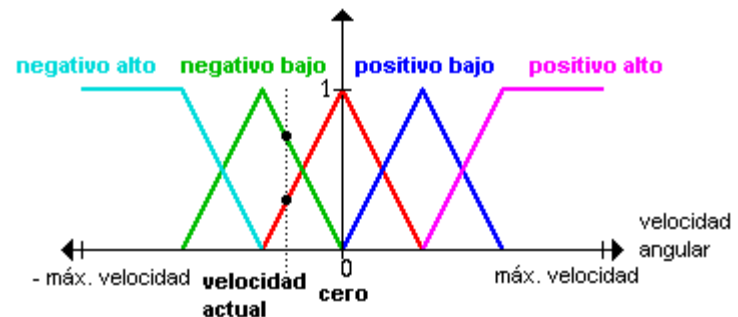
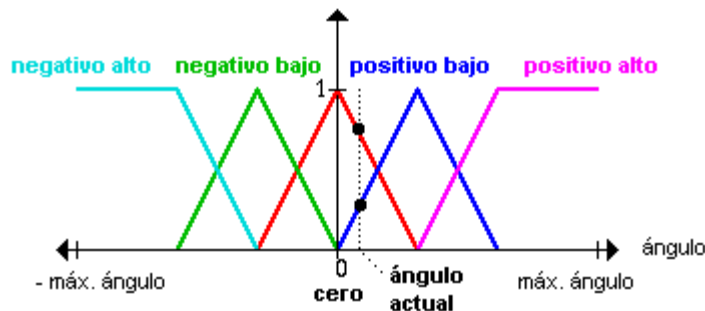
- Lo mismo se hace para el ángulo entre la plataforma y la pértiga, y la velocidad angular de este ángulo:



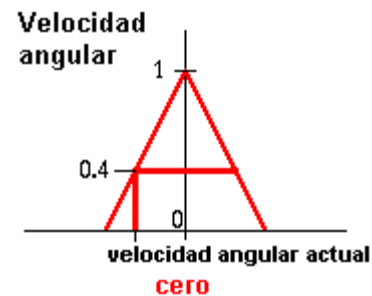
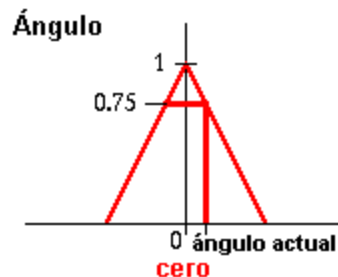
- Algunas reglas del sistema difuso
 - Si el ángulo es cero **y** la velocidad angular es cero **entonces** la velocidad será cero.
 - Si el ángulo es cero **y** la velocidad angular es positiva baja **entonces** la velocidad será positiva baja.

Ejemplo

- Consideremos un valor actual para el ángulo y velocidad angular

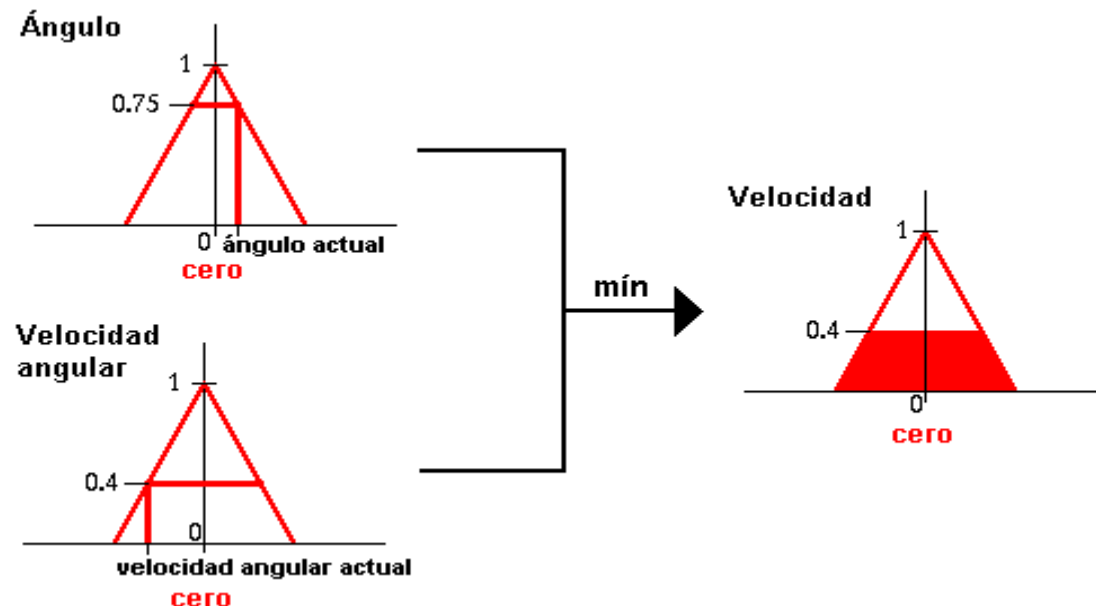


- Ahora mostraremos como aplicar nuestras reglas a esta situación real. Veamos como aplicar la regla
 - Si el ángulo es cero y la velocidad angular es cero **entonces** la velocidad será cero.



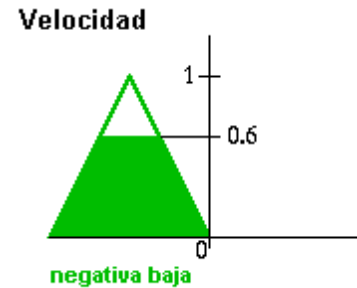
Ejemplo

- Como las dos partes de la condición de nuestra regla están unidas por una Y (operación lógica AND) calculamos el $\min(0.75, 0.4) = 0.4$ y cortamos el conjunto borroso "cero" de la variable "velocidad" a este nivel (según nuestra regla):

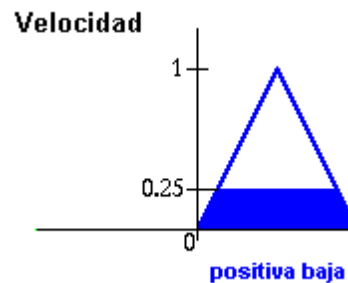


Ejemplo

- Por su parte, el resultado de las reglas
 - Si el ángulo es cero y la velocidad angular es negativa baja **entonces** la velocidad será negativa baja

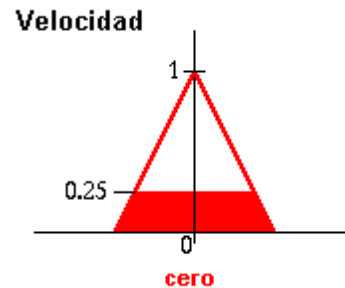


- Si el ángulo es cero y la velocidad angular es positiva baja **entonces** la velocidad será positiva baja

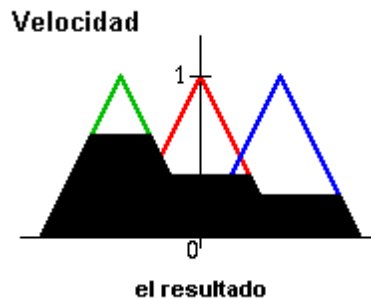


Ejemplo

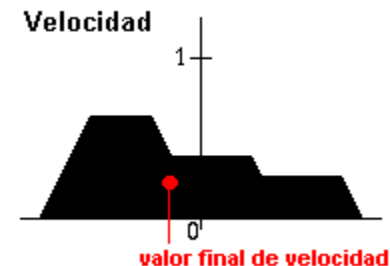
- Si el ángulo es positivo bajo y la velocidad angular es negativa baja entonces la velocidad será cero



- Estas cuatro reglas solapadas desembocan en un resultado único:



Tenemos que escoger un valor representativo como salida final usando alguno de los métodos de defuzzification



CASO DE ESTUDIO 2

SISTEMA CLASIFICADOR DE LIRIOS

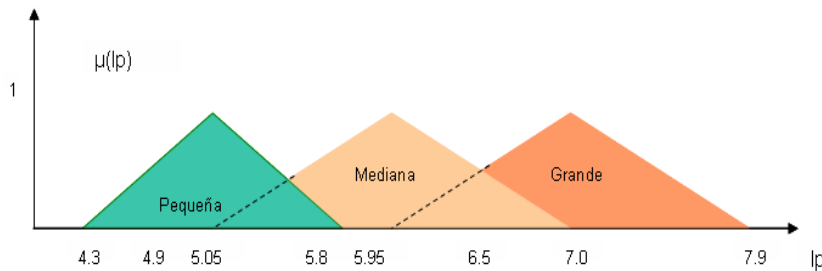
- Clasificar correctamente las variedades de flor Iris (Lirios), respecto a las clases: setosa, virgínica y versicolor.
- Los atributos utilizados para la clasificación son la longitud y ancho de los pétalos, y el largo y ancho de los tallos.

Variables

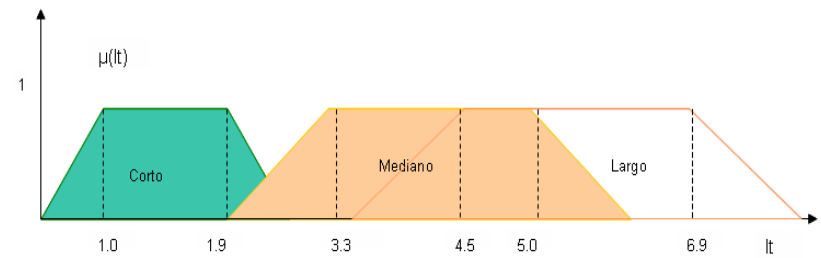
- Variables de entrada: ancho del pétalo (ap), longitud del pétalo (lp), ancho del tallo (at) y longitud del tallo (lt).
- Variables de salida: Clase.

CASO DE ESTUDIO 2

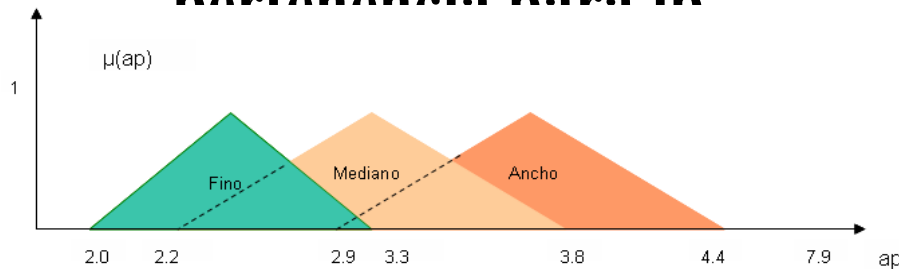
SISTEMA CLASIFICADOR DE LIRIOS



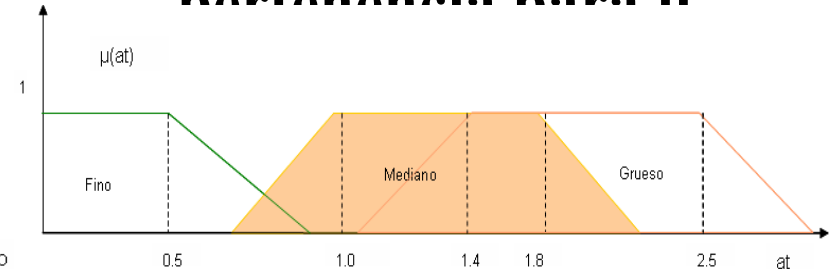
Funciones de pertenencia para lp



Funciones de pertenencia para lt



Funciones de pertenencia para ap



Funciones de pertenencia para at

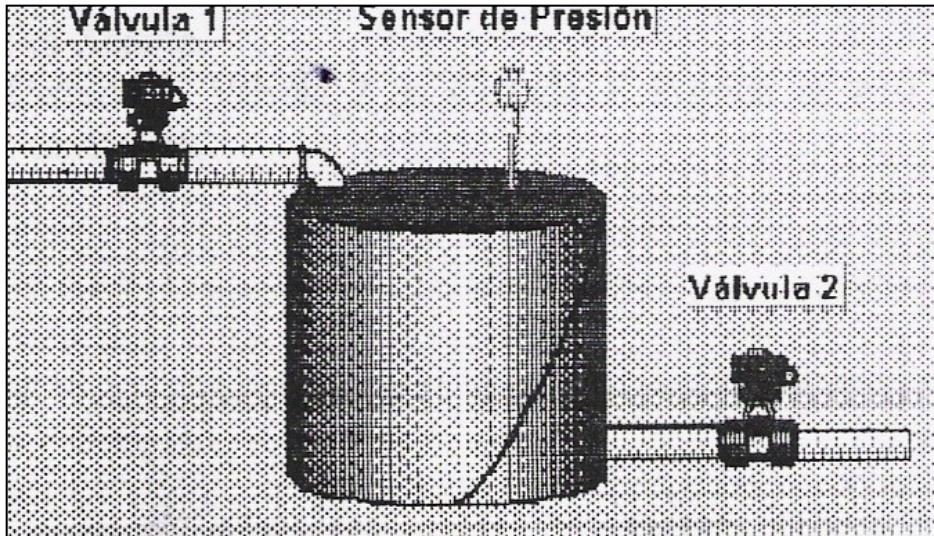
CASO DE ESTUDIO 2

SISTEMA CLASIFICADOR DE LIRIOS

1. Si lp es Pequeña Y ap es Ancho Y lt es Corto Y at es Fino Entonces Clase es Setosa.
2. Si lp es Mediano Y ap es Fino Y lt es Mediano Y at es Mediano Entonces Clase es Virgínica.
3. Si lp es Grande Y ap es Mediano Y lt es Largo Y at es Largo Entonces Clase es Versicolor.
4. Si lp es Grande Y ap es Fino Y lt es Largo Y at es Grande Entonces Clase es Versicolor.
5. Si lp es Mediano Y ap es Mediano Y lt es Mediano Y at es Mediano Entonces Clase es Virginica.
6. Si lp es Pequeña Y ap es Mediano Y lt es Fino Y at es Fino Entonces Clase es Setosa.

CASOS DE ESTUDIO 3

SISTEMA DE CONTROL PARA UN TANQUE



Las variables en el sistema son:

Variables de entrada:

presión, válvula 1 y válvula 2.

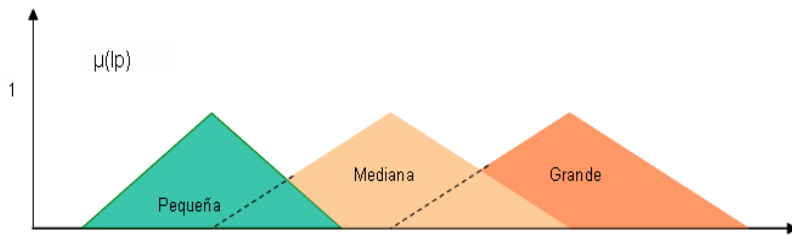
Variables de salida:

válvula 1 y válvula 2.

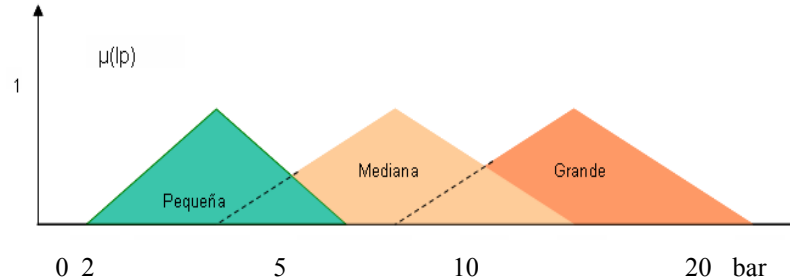
Válvula 1 y Válvula 2:

tipo Enumerado, con valores difusos de apertura: mediana, grande, pequeña.

CASOS DE ESTUDIO 3



**Funciones de pertenencia
para válvulas**



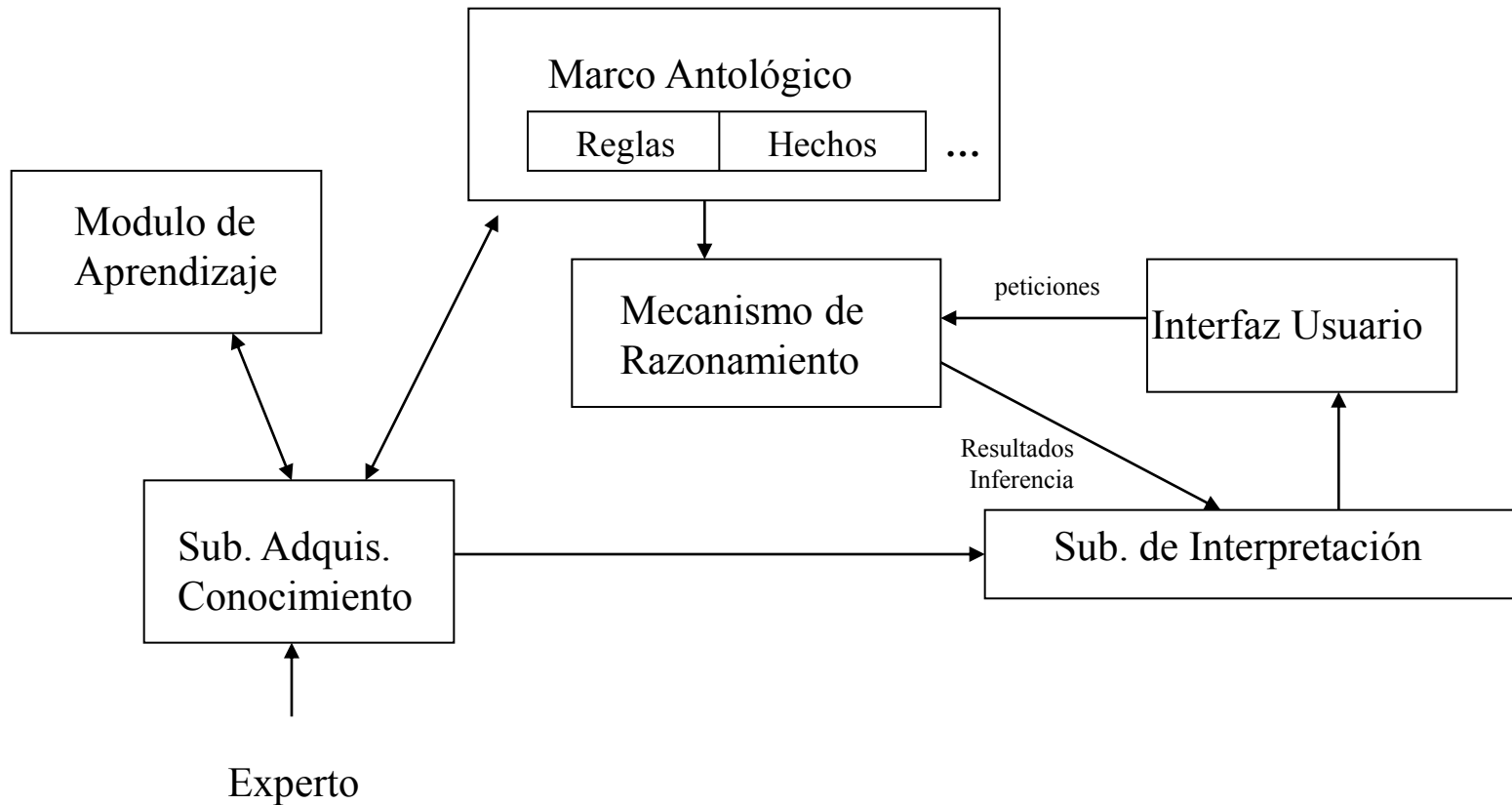
**Funciones de pertenencia
para presión**

CASOS DE ESTUDIO 3

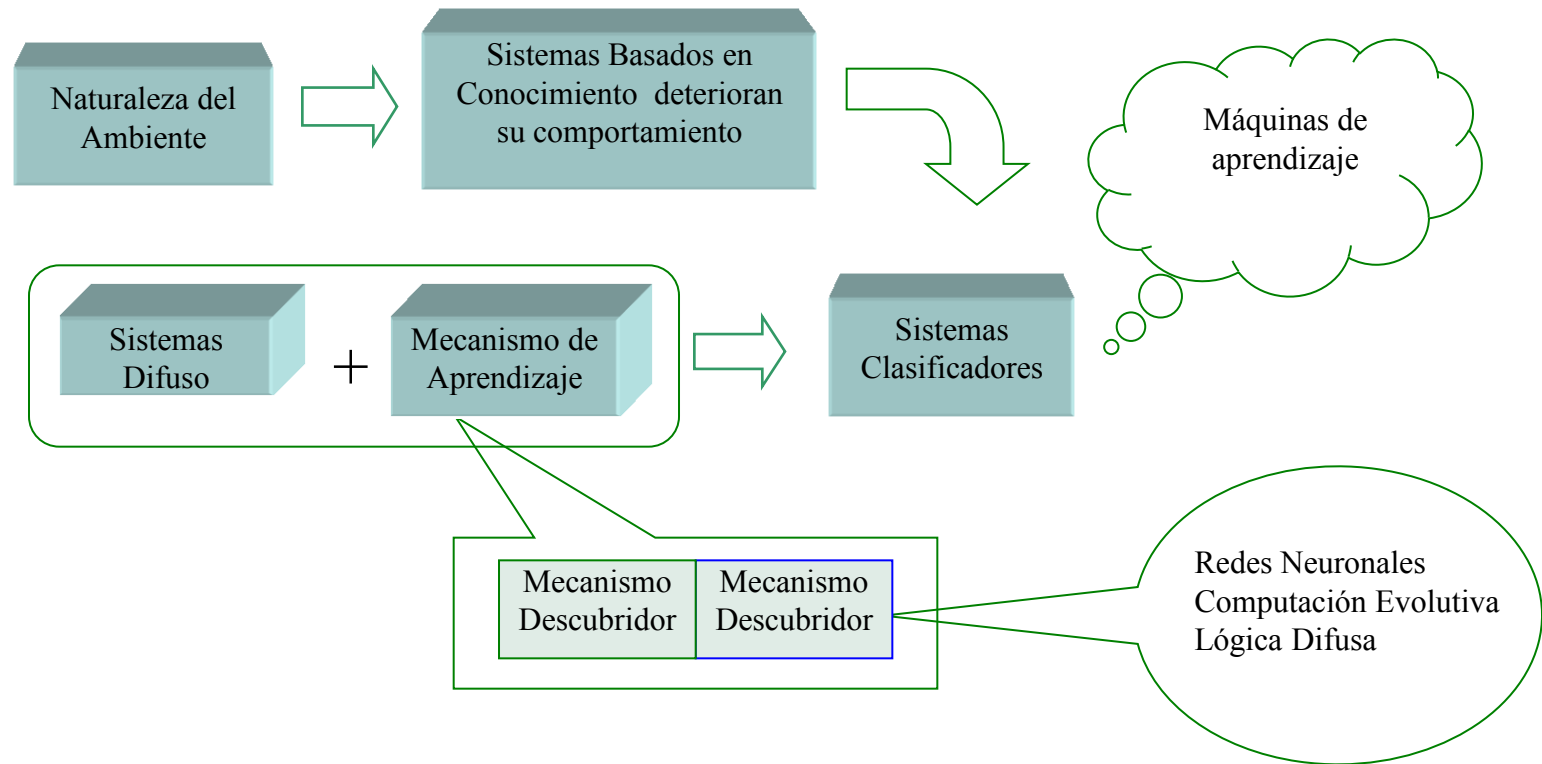
SISTEMA DE CONTROL PARA UN TANQUE

1. Si Valvula1 == grande Y Valvula2 == grande Y Presión < 5 Entonces Valvula1 = mediana Y Valvula2 = pequeña.
2. Si Valvula1 == mediana Y Valvula2 == grande Y Presión > 10 Entonces Valvula1 =pequeña Y Valvula2 = grande.
3. Si Valvula1 == grande Y Valvula2 ==mediana Y Presión > 10 Entonces Valvula1 =pequeña Y Valvula2 = mediana.
4. Si Valvula1 == grande Y Valvula2 == mediana Y Presión >= 5 Y Presión <=10 Entonces Valvula1 = grande Y Valvula2 = pequeña.
5. Si Valvula1 == pequeña Y Valvula2 == grande Y Presión < 5 Entonces Valvula1 = grande Y Valvula2 = pequeña.
6. Si Valvula1 == pequeña Y Valvula2 == grande Y Presión < 5 Entonces Valvula1 = grande Y Valvula2 = pequeña.

Sistemas Basados en Conocimiento

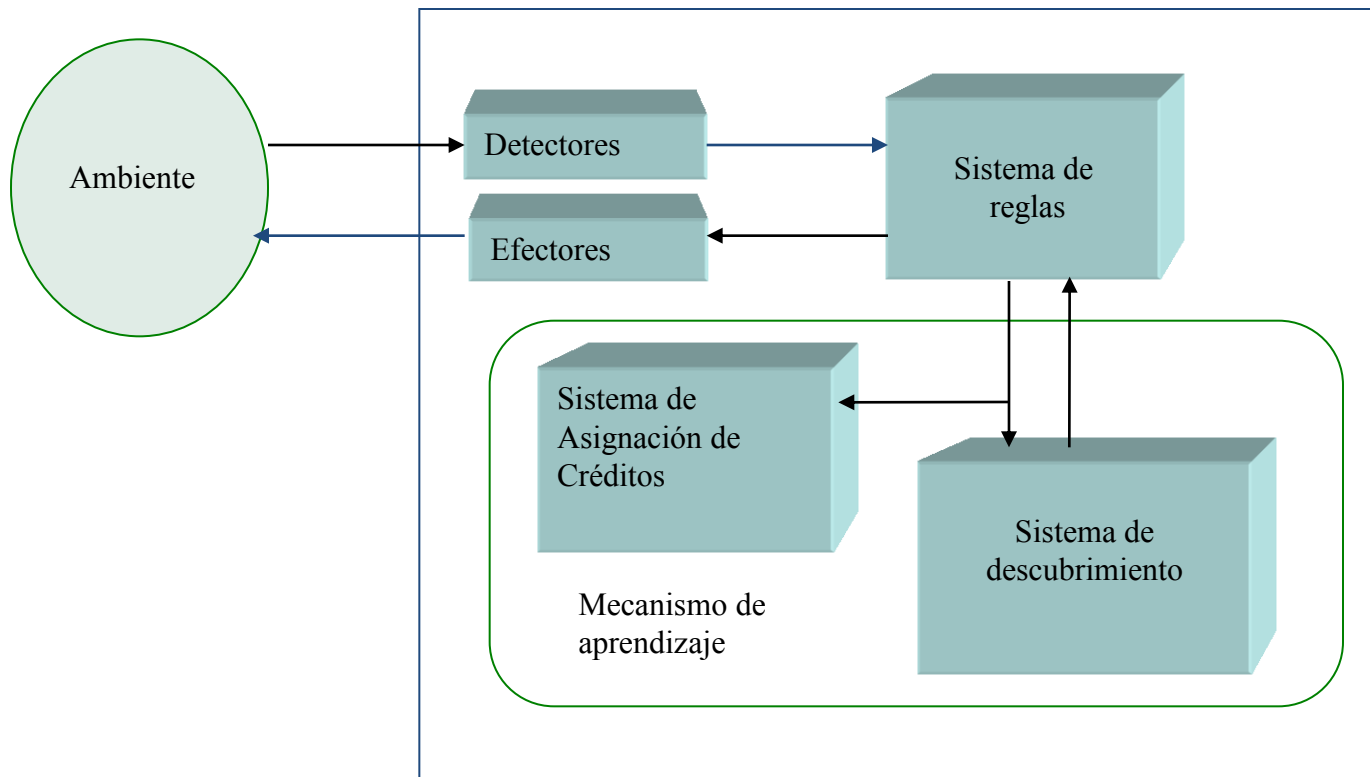


Máquinas de Aprendizaje



SISTEMAS CLASIFICADORES

Aprendizaje



Aprendizaje

