Daniel Felipe Ladino Velásquez

Jonatan Hernández Henao

OCTUBRE DE 2020

PERCEPTRÓN Y LÓGICA DIFUSA: Computación Blanda

UTP | Pereira

2020

# CONTENIDO

[1 CONTENIDO 1](#_Toc54775578)

[2 PRESENTACIÓN 2](#_Toc54775579)

[3 EL PERCEPTRÓN 5](#_Toc54775580)

[4 LÓGICA DIFUSA - INTRODUCCIÓN 9](#_Toc54775581)

[5 CONCLUSIONES 12](#_Toc54775582)

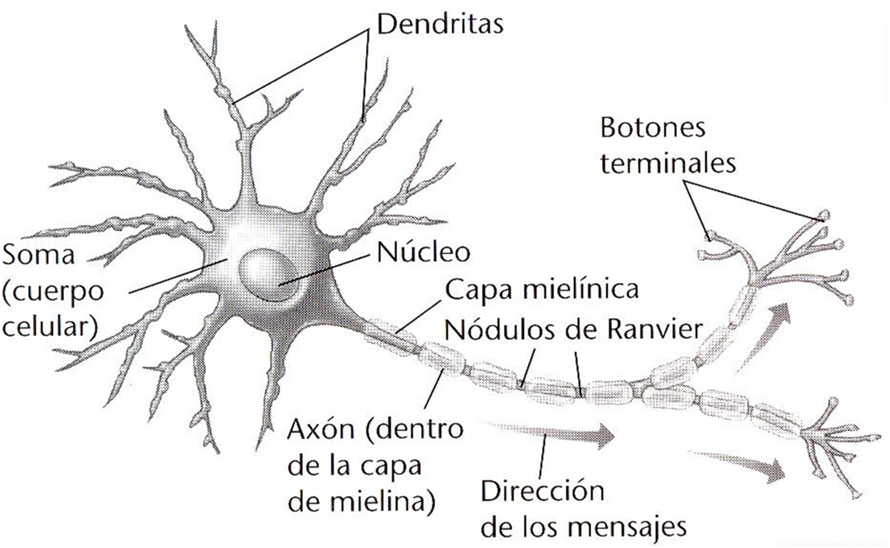
[6 BIBLIOGRAFÍA 13](#_Toc54775583)

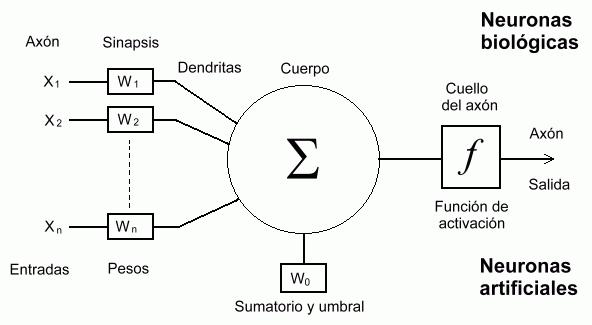
# PRESENTACIÓN

La presente monografía está orientada a la descripción de los elementos básicos de las neuronas artificiales, en particular el perceptrón, y la teoría fundamental de la lógica difusa.

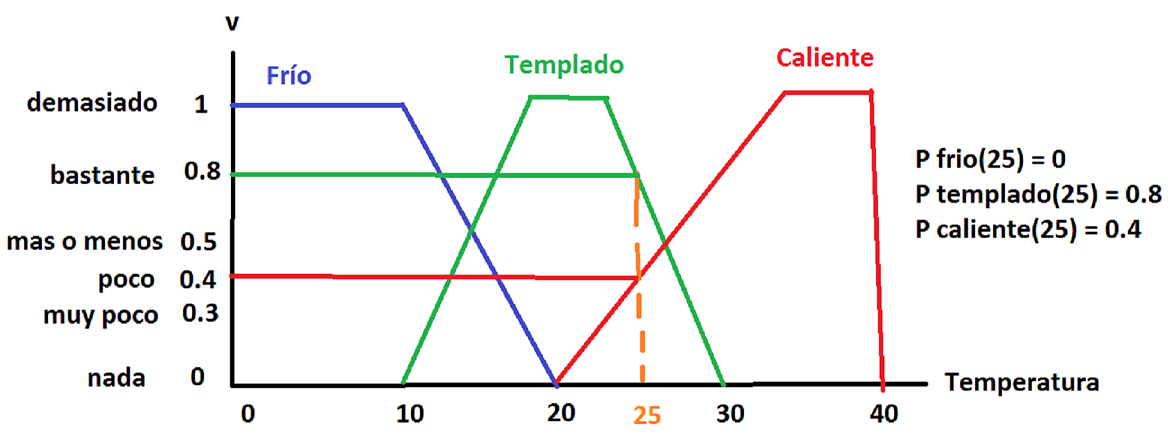
En el documento se analizan los diferentes elementos que componen ambas tecnologías, mostrando las relaciones matemáticas que dan soporte a las funcionalidades tanto del perceptrón como a los factores de incertidumbre que dan sentido a la lógica difusa.

A grandes rasgos, las redes neuronales se basan en los modelos que subyacen a las redes neuronales biológicas. El siguiente diagrama adelante algunos elementos presentes en esta tecnología.





La lógica difusa se basa en la concepción de que la verdad (y la falsedad) no son absolutas. Por este motivo, todos los conceptos que concibe el ser humano tienen cierto grado de certeza, el cual se expresa fácilmente si recurrimos a un esquema como el que se ve a continuación.



En este esquema se afirma que el Frío, la sensación de Templado, y algo que es Caliente, son curvas que varían de acuerdo con la temperatura, según se ve. En el caso particular de tener una temperatura ambiente de 25 grados, dicha temperatura tendrá un valor de verdad respecto de “Caliente” de sólo 0.4. En cambio, los 25 grados representarán, en la curva de “Templado”, un valor de verdad de 0.8. Se aprecia, además, que dichos valores se relacionan, de manera bastante cercana, con frases y/o palabras que utiliza el ser humano para describir situaciones de la vida real.

En las próximas secciones se verán estas tecnologías con un mayor grado de detalle.

**AUTOR:**

**Daniel Felipe Ladino Velásquez**

**Jonatan Hernández Henao**

**<CODIGO>**

**1092915812**

**1053864927**

**<CORREO Estudiante>**

**d.ladino@utp.edu.co**

**jonatan.hernandez@utp.edu.co**

**<GITHUB Estudiante>**

**https://github.com/Danielladino**

**https://github.com/jonatanhh**

# EL PERCEPTRÓN

A continuación, se presenta el algoritmo básico para la conversión numérica basada en divisiones sucesivas.

Es la neurona artificial o una unidad básica de inferencia en forma de discriminador lineal, a partir del cual se obtiene o desarrolla un algoritmo con la capacidad de generar un criterio para seleccionar un pequeño grupo a partir de un grupo de componentes más grande.

Este algoritmo tiene una limitación y es que si se dibuja un grafico estos elementos se deben poder separar con un hiperplano únicamente los elementos "deseados" discriminándolos (separándolos) de los "no deseados".

El perceptrón puede utilizarse con otros tipos de perceptrones o de neurona artificial, para formar una red neuronal artificial más compleja.

El perceptrón es la forma más simple de una red neuronal usada para la clasificación de un tipo especial de patrones, los linealmente separables (es decir, patrones que se encuentran a ambos lados de un hiperplano). Básicamente, consiste de una neurona con pesos sinápticos y umbral ajustables, como se muestra en la figura. El algoritmo usado para ajustar los parámetros libres de esta red neuronal apareció por primera vez en un procedimiento de aprendizaje desarrollado por Rosenblatt (1958) para su modelo de perceptrón del cerebro. En realidad, Rosenblatt demostró que, si los patrones usados para entrenar el perceptrón son sacados de dos clases linealmente separables, entonces el algoritmo del perceptrón converge y toma como superficie de decisión un hiperplano entre estas dos clases. La prueba de convergencia del algoritmo es conocida como el teorema de convergencia del perceptrón.

El perceptrón de una capa descrito en la figura tiene sólo una neurona. Dicho perceptrón está limitado a realizar clasificación de patrones con sólo dos clases. Expandiendo la capa de salida del perceptrón para incluir más que una neurona, podemos realizar dicha clasificación con más de dos clases. Sin embargo, las clases tendrían que ser linealmente separables para que el perceptrón trabaje correctamente.

El modelo biológico más simple de un perceptrón es una neurona y viceversa. Es decir, el modelo matemático más simple de una neurona es un perceptrón. La neurona es una célula especializada y caracterizada por poseer una cantidad indefinida de canales de entrada llamados dendritas y un canal de salida llamado axón. Las dendritas operan como sensores que recogen información de la región donde se hallan y la derivan hacia el cuerpo de la neurona que reacciona mediante una sinapsis que envía una respuesta hacia el cerebro, esto en el caso de los seres vivos.

Una neurona sola y aislada carece de razón de ser. Su labor especializada se torna valiosa en la medida en que se asocia a otras neuronas, formando una red. Normalmente, el axón de una neurona entrega su información como "señal de entrada" a una dendrita de otra neurona y así sucesivamente. El perceptrón que capta la señal en adelante se extiende formando una red de neuronas, sean éstas biológicas o de sustrato semiconductor (compuertas lógicas).

El perceptrón usa una matriz para representar las redes neuronales y es un discriminador terciario que traza su entrada x (un vector binario) a un único valor de salida f(x) (un solo valor binario) a través de dicha matriz.

C:\Users\ladin\Desktop\Sin título1.png

Donde w es un vector de pesos reales y w.x es el producto escalar (que computa una suma ponderada). u es el 'umbral', el cual representa el grado de inhibición de la neurona, es un término constante que no depende del valor que tome la entrada.

El valor de f(x) (0 o 1) se usa para clasificar x como un caso positivo o un caso negativo, en el caso de un problema de clasificación binario. El umbral puede entenderse como una manera de compensar la función de activación, o una forma de fijar un nivel mínimo de actividad a la neurona para considerarse como activa. La suma ponderada de las entradas debe producir un valor mayor que u para cambiar la neurona de estado 0 a 1.

En el perceptrón, existen dos tipos de aprendizaje, el primero utiliza una tasa de aprendizaje mientras que el segundo no la utiliza. Esta tasa de aprendizaje amortigua el cambio de los valores de los pesos.

El algoritmo de aprendizaje es el mismo para todas las neuronas, todo lo que sigue se aplica a una sola neurona en el aislamiento. Se definen algunas variables primero:

x(j) denota el elemento en la posición j en el vector de la entrada

w(j) el elemento en la posición j en el vector de peso

y denota la salida de la neurona

\delta denota la salida esperada

\alpha es una constante tal que 0<\alpha <1

Los dos tipos de aprendizaje difieren en este paso. Para el primer tipo de aprendizaje, utilizando tasa de aprendizaje, utilizaremos la siguiente regla de actualización de los pesos:

C:\Users\ladin\Desktop\Sin título2.png

Para el segundo tipo de aprendizaje, sin utilizar tasa de aprendizaje, la regla de actualización de los pesos será la siguiente:

C:\Users\ladin\Desktop\Sin título3.png

Por lo cual, el aprendizaje es modelado como la actualización del vector de peso después de cada iteración, lo cual sólo tendrá lugar si la salida y difiere de la salida deseada \delta. Para considerar una neurona al interactuar en múltiples iteraciones debemos definir algunas variables más:

x{i} denota el vector de entrada para la iteración i

w{i} denota el vector de peso para la iteración i

y{i} denota la salida para la iteración i

 denota un periodo de aprendizaje de m iteraciones

En cada iteración el vector de peso es actualizado como sigue:

Para cada pareja ordenada

Pasar a la regla de actualización

El periodo de aprendizaje se dice que es separable linealmente si existe un valor positivo  y un vector de peso w tal que: para todos los i.

Novikoff (1962) probó que el algoritmo de aprendizaje converge después de un número finito de iteraciones si los datos son separables linealmente y el número de errores está limitado a: .

Sin embargo, si los datos no son separables linealmente, la línea de algoritmo anterior no se garantiza que converja.

# LÓGICA DIFUSA - INTRODUCCIÓN

La teoría base de la lógica difusa se presenta a continuación.

En los años sesenta el ingeniero lofty A. zadeh investiga y explora por primera vez la lógica difusa lo cual el llamo el principio de la incompatibilidad: “conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre el comportamiento disminuye el umbral más allá del cual, la precisión y el significado son características excluyentes”

La lógica difusa nos permite representar el conocimiento común por medio de un lenguaje matemático enfocado en la teoría de conjuntos difusos y funciones asociadas a ellos, podemos trabajarlo tanto por los datos numéricos como términos lingüísticos, pero cabe resaltar y aclarar que los son mucho más confiables y precisos los datos numéricos.

La lógica difusa tiene la capacidad de reproducir aceptablemente los modos usuales del razonamiento, se puede decir que la lógica difusa es la ciencia del razonamiento la cual se refiere a los principios formales del razonamiento difuso.

La lógica difusa tiene como características primordiales la flexibilidad, la tolerancia con la imprecisión, la capacidad para moldear y mejorar problemas no lineales, y su base en el lenguaje natural.

Aristóteles y otros filósofos griegos consideraban que existían ciertos grados de veracidad y falsedad, pero platón trabajo con grados de pertenencia.

Al comienzo la lógica difusa tenía un gran impacto en la comunidad científica, donde investigadores se volvieron fieles seguidores de las teorías de zadeh, mientras este autor seguía indagando para ampliar los fundamentos de la teoría de conjuntos difusos los investigadores fueron buscando nuevas teorías durante la siguiente década después de su aparición, durante la primera década gran parte de estructuras lógicas y matemáticas son generalizadas en términos de lógica difusa como: relaciones lógicas, funciones, grupos, operaciones, operadores, algoritmos, etc…

En 1974 assilian y mamdani establecen un aporte importante en el desarrollo de la lógica difusa el cual consistía en desarrollar el primer controlador difuso para una máquina de vapor, pero la primera implantación real de un controlador de este tipo fue por F.L. smidth en una planta cementera en Dinamarca

En 1983 Fuji aplica la lógica difusa para el control de inyección química en plantas depuradoras de agua por primera vez en Japón y en 1987 Hitachi pone en marcha un controlador fuzzy para el control del tren-metro de Sendai, y la empresa Omron desarrolla los primeros controladores difusos comerciales.

Paralelamente al desarrollo de las aplicaciones de la lógica difusa, investigadores teóricos siguen, en la década de los ochenta, el camino iniciado por Mamdani. Así, Takagi y Sugeno desarrollan la primera aproximación para construir reglas fuzzy a partir de datos de entrenamiento, y aunque en un principio no tiene mucha repercusión, más tarde será el punto de partida para investigar la identificación de modelos fuzzy.

Otro de los factores que contribuye a seguir con la investigación en este campo es el creciente interés en las redes neuronales y su similitud con los sistemas fuzzy; la tendencia es buscar vías de

relación entre las dos técnicas y los resultados son los llamados neuro-fuzzy systems, sistemas fuzzy que usan métodos de aprendizaje basados en redes neuronales para identificar y optimizar sus parámetros. B. Kosko es conocido por su contribución a los sistemas neuro-fuzzy y con sus publicaciones introdujo en la lógica difusa a muchos lectores interesados en las redes neuronales.

En la década de los noventa, además de las redes neuronales y los sistemas fuzzy, hacen su aparición los algoritmos genéticos. Estas tres técnicas computacionales, que pueden combinarse de múltiples maneras y se pueden considerar complementarias, son herramientas de trabajo muy potentes en el campo de los sistemas de control en la última década.

En realidad, la intención original del profesor Zadeh era crear un formalismo para manipular de forma más eficiente la imprecisión y la vaguedad del razonamiento humano expresado lingüísticamente, sin embargo, causó cierta sorpresa que el éxito de la lógica borrosa llegase en el campo del control automático de procesos. Esto se debió básicamente al boom que la lógica borrosa causó en Japón, iniciado en 1987 y que alcanzó su máximo apogeo a principios de los noventa. Este boom fue el resultado de una estrecha colaboración entre el gobierno, las universidades y las industrias japonesas, estableciéndose dos proyectos nacionales a gran escala llevados a cabo por el Ministerio de Industria y Comercio (MITI) y la Agencia de Ciencia y Tecnología (STA) en consorcio con el LIFE, Laboratory for International Fuzzy Research, y en los que se involucraron más de 50 compañías durante seis años.

Desde entonces, han sido infinidad los productos lanzados al mercado que usan tecnología borrosa, muchos de ellos utilizando la etiqueta fuzzy como símbolo de calidad y prestaciones avanzadas. El control difuso ha sido aplicado con éxito en muy diversas ramas

tecnológicas, por ejemplo la metalurgia, robots para la fabricación, controles de maniobras de aviones, sensores de imagen y sonido (sistema de estabilización de la imagen en cámaras fotográfica y de video Sony, Sanyo y Cannon), lavadoras (Panasonic y Bosch) que son capaces de autorregular la cantidad de jabón que requiere un lavado dependiendo del grado de suciedad de la ropa, aire acondicionado (Mitsubishi) en el que el sistema fuzzy evita las oscilaciones entre el exceso y el defecto de temperatura), rice-cooker capaces de elaborar diversas variedades de arroz regulando la cantidad de agua y la temperatura en cada caso para que el grano quede cocido y suelto, en automoción, sistemas de frenado ABS (Mazda y Nissan), cambio automático de Renault, control automático de velocidad que controla la frenada en casos peligrosos y selecciona la relación de marchas a partir del rendimiento del motor, climatizadores, fotocopiadoras (ajusta el voltaje del tambor a partir de la densidad de la imagen, la temperatura y la humedad), lavaplatos (ajusta el ciclo de lavado y enjuague a partir del número de platos y cantidad de comida adherida), ascensores (reduce el tiempo de espera a partir del número de personas), humidificadores (ajusta el contenido de humedad a las condiciones de la habitación), mejoras en imágenes médicas (ajustando el contraste en los bordes), sistemas de reconocimiento de escritura, hornos microondas (establece y afina el programa de energía y cocción), neveras (establece los tiempos de descongelación y enfriamiento en función del uso que se haga), televisores (ajusta el color de la pantalla y la textura de cada imagen), mecanismos de atraque automático de naves espaciales, sistemas automáticos de regulación de la cantidad de anestesia que se suministra a los pacientes en un quirófano aunque bajo supervisión médica, por supuesto-, sistemas de concesión o denegación- automática de créditos según el perfil económico del solicitante, etc...

Estas son algunas de las muchísimas aplicaciones de la lógica difusa, que ya están funcionando en el campo de los llamados sistemas expertos. Todos estos sistemas utilizan información, esencialmente, imprecisa con el fin de lograr sus cometidos.

# CONCLUSIONES

El desarrollo de las temáticas elaboradas en clase utilizando el lenguaje JavaScript prueba ser un mecanismo de gran valor para el aprendizaje de los conceptos básicos de la materia.

# BIBLIOGRAFÍA

<https://repl.it>

<https://es.wikipedia.org/wiki/Perceptr%C3%B3n>

<http://bibing.us.es/proyectos/abreproy/11084/fichero/Memoria+por+cap%C3%ADtulos+%252FCap%C3%ADtulo+4.pdf>+