

El perceptrón, su entrenamiento, backpropagation, y lógica difusa

Presentado por:

- Juan Camilo Olarte Betancourt
- Carlos Daniel Morales Isaza

El Perceptrón

""El perceptrón simple es una red neuronal artificial de una sola capa, inspirada en el comportamiento y funcionamiento del cerebro humano". Para empezar, el perceptrón simple es la red neuronal artificial más antigua. En el campo de la inteligencia artificial, el perceptrón simple es una red neuronal artificial de una sola capa, inspirada en el comportamiento y funcionamiento del cerebro humano. Una neurona cerebral posee el aspecto y las partes que se muestran en la figura

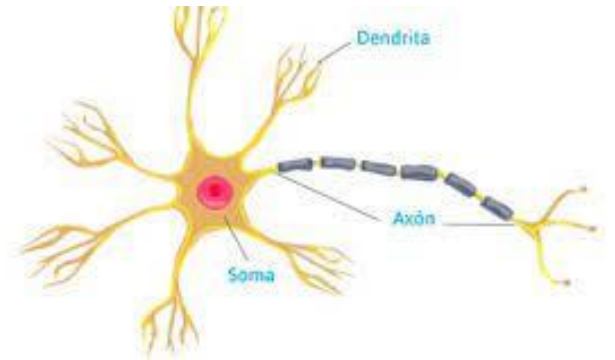


Figura 1. Neurona y sus partes.

El Perceptrón: modelo

En la figura se muestra el modelo del perceptrón, en este se observa cómo de forma similar a lo que pasa con una neurona el perceptrón recibe señales enviadas por otros perceptrones (x_1 a x_5), esas señales de entrada son valores numéricos que representan lo que será procesado. Los valores w_1 a w_5 representan los pesos sinápticos en las dendritas de una neurona, lo que hace cada peso sináptico es simplemente multiplicar a su entrada correspondiente y define la importancia relativa de cada entrada.

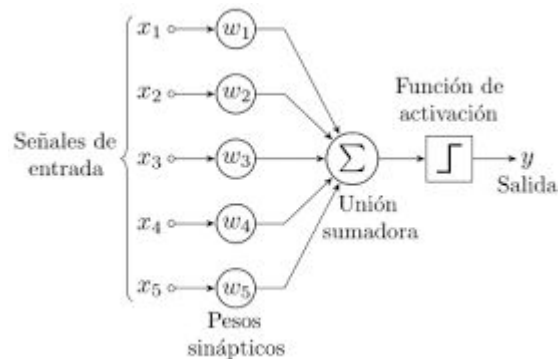


Figura 2. El perceptrón

El Perceptrón:

Función de entrada

Los perceptrones convierten sus múltiples entradas en una entrada global. Para combinar este montón de entradas se utiliza una función de entrada. Se emplean variedades de funciones de entrada según sean apropiadas para el caso de estudio, algunas de las funciones de entradas más comúnmente utilizadas y conocidas son:

- 1) Sumatorio de las entradas pesadas: es la suma de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos.
- 2) Multiplicatorio de las entradas pesadas: es el producto de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos.
- 3) Máximo de las entradas pesadas: solamente toma en consideración el valor de entrada más fuerte, previamente multiplicado por su peso correspondiente

El Perceptrón:

Función de activación y función de salida

El funcionamiento de un perceptrón inspirado en una neurona consiste en que si la suma de las entradas excitatorias supera el umbral de activación, la neurona se activa y emite una respuesta (equivalente al valor de 1), en caso contrario no se activará y no emitirá respuesta (representado por el valor de 0). Lo que se hace es aplicar una función de activación, que puede ser, por ejemplo, una función tipo escalón o sigmoidea.

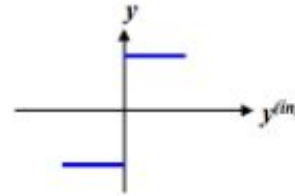


Figura 3. Escalón

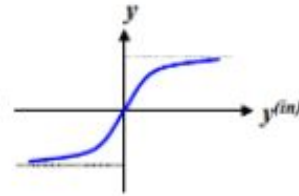


Figura 4. Sigmoidea

El último componente del perceptrón es la función de salida, la función de salida determina que valor se transfiere a los perceptrones vinculados.

El Perceptrón: Perceptrón multicapa

Un perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y una o más capas ocultas; aunque se ha demostrado que para la mayoría de problemas bastará con una sola capa oculta. En la figura podemos observar un perceptrón multicapa formado por una capa de entrada, una capa oculta y una de salida.

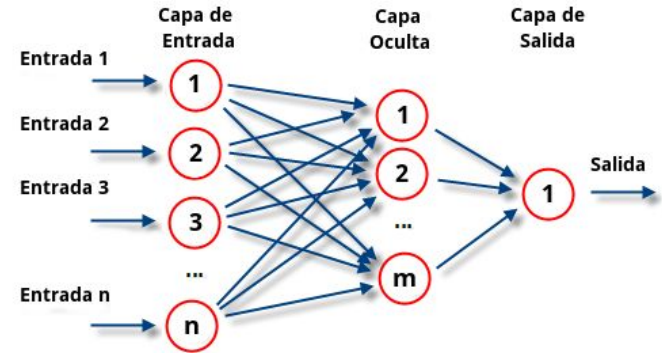


Figura 5. Perceptrón multicapa.

Backpropagation

A decorative graphic in the top right corner consisting of several overlapping circles in various shades of gray, creating a modern, abstract background element.

La propagación hacia atrás de errores o retropropagación (del inglés backpropagation) es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales.

Es un algoritmo ampliamente utilizado para entrenar redes neuronales, Existen generalizaciones de Backpropagation para otras redes neuronales artificiales y para funciones en general. Todas estas clases de algoritmos se denominan genéricamente " Backpropagation".

Backpropagation: Funcionamiento

El algoritmo tiene dos partes, en la primera, cada grupo o vector de entradas es presentado a la red, y está, usualmente inicializada con pesos aleatorios, genera algunas salidas. Estas salidas de la red son comparadas con las salidas que se espera que la red asocie a esas entradas, y se calcula la diferencia o error.

Backpropagation: Funcionamiento (continuación)

Se presenta el segundo vector de entrada, también conocido como ejemplo; nuevamente se propaga la red, y se evalúa el error entre la salida de la red y las salidas que se esperarían. El error debido a todos los ejemplos es acumulado, y se repite esto con todos los ejemplos que incluya el conjunto de datos que se tenga.

Backpropagation: Funcionamiento (continuación)

En la segunda parte, con base en el error acumulado, se empiezan a cambiar los pesos de cada conexión en la capa de salida, de manera que el error disminuya.

Se utiliza el método del descenso del gradiente para lograr el mínimo valor posible de error cuadrático medio. Con esto se modifican los pesos sinápticos hasta que el sistema evidencie una conducta deseable. Este método de aprendizaje es de tipo no supervisado

Lógica difusa

A decorative graphic in the top right corner consisting of several overlapping circles in various shades of gray, creating a modern, abstract background element.

Es la lógica que utiliza expresiones que no son ni totalmente ciertas ni completamente falsas, es decir, es la lógica aplicada a conceptos que pueden tomar un valor cualquiera dentro de un conjunto de valores que oscilan entre dos extremos, la verdad absoluta y la falsedad total.

Lógica difusa:

variables lingüísticas

Una Variable Lingüística es aquella variable cuyos valores o estados son expresados por palabras. Por ejemplo, “Velocidad” es una variable lingüística, si sus valores son “muy lento”, “lento”, “rápido” y “muy rápido” (véase en la Figura 1.9), siendo éstos valores denominados como etiquetas lingüísticas.

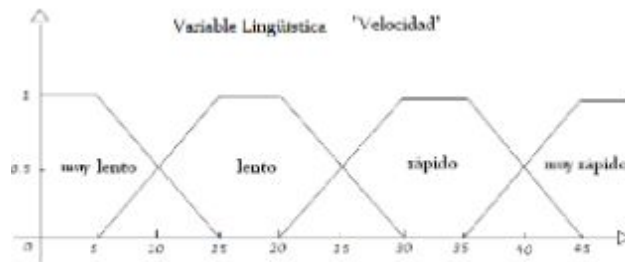


Figura 1.9: Valores lingüísticos de velocidad

Lógica difusa:

Modelado difuso de sistemas

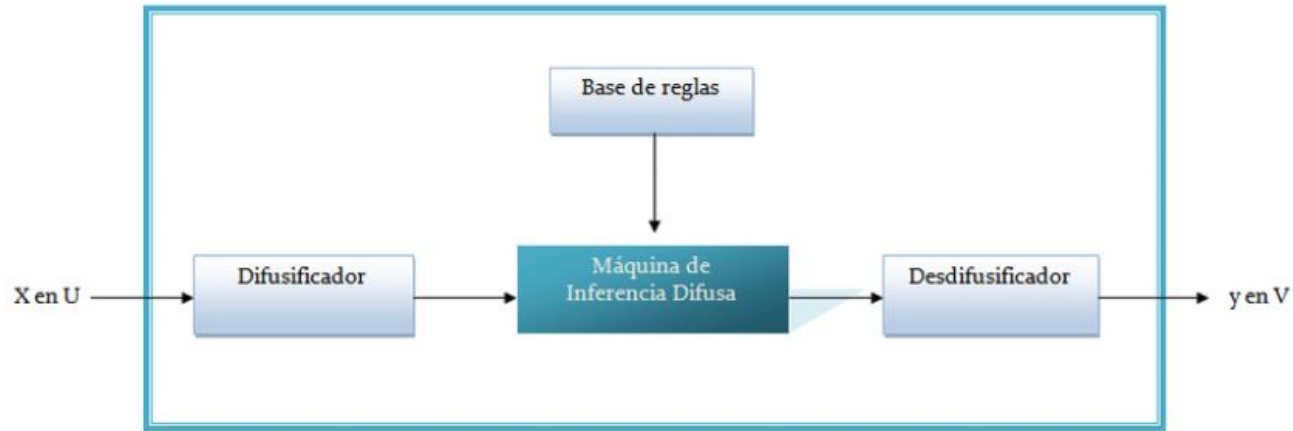


Figura 6. Sistema difuso con difusificador y desdifusificador

Lógica difusa:

Modelado difuso de sistemas

Secciones que constituyen el modelado:

Difusificación es la conversión de valores del mundo real al dominio difuso mediante el uso de funciones de membresía. Después de efectuar la difusificación, es necesario tener una base de conocimiento, para posteriormente determinar la **máquina de inferencia o proceso de inferencia difuso** que es la parte más importante de todo modelo difuso, ya que es la parte que efectuará todo el proceso en base a la información disponible. Luego de evaluar las reglas difusas, que relacionan las variables difusas, es generalmente necesario trasladar los valores resultantes a su mundo original, el real, a esto se le conoce como **desdifusificación**.

Código en COLAB sobre lógica difusa

El problema de lógica difusa que se desarrolla es la detección de la probabilidad de accidentalidad vehicular teniendo en cuenta dos factores: la experticia del conductor al volante y el estado del clima(la lluviosidad). Estas variables lingüísticas tienen los valores o estados poor, average y good.

Se emplea la función de membresía triangular para convertir los valores de experiencia, lluviosidad y prob_accidente al dominio difuso. El proceso de inferencia determina la salida más adecuada del sistema a partir de la regla difusa, el conocimiento(las funciones de membresía triangular) y de los datos de entrada. Todo esto se puede observar en el código adjunto en github.

Código en COLAB sobre lógica difusa

Caso de prueba:

En una escala de 0 a 10, un conductor con experiencia con valor 2 y en un clima (lluviosidad) con valor 10, el porcentaje de accidentalidad es del 77.7% como se muestra en la figura.



Porcentaje de accidentalidad:
77.71428571428565 %

