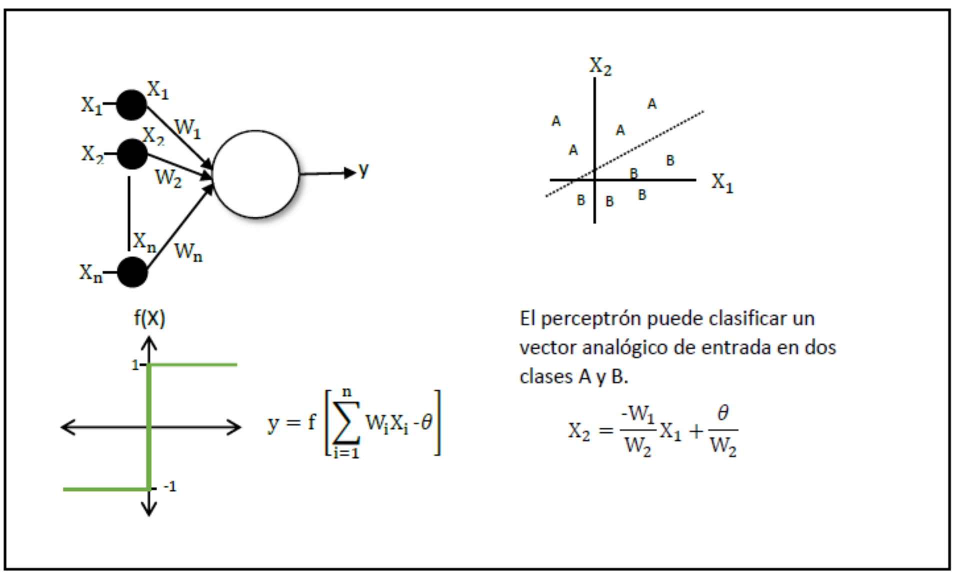
REDES NEURONALES CON CONEXIONES HACIA ADELANTE: EL PERCEPTRÓN

Estas redes son buenos clasificadores de patrones y utilizan aprendizaje supervisado.



$$f(x) = \begin{cases} 1, \text{si} & x \geq 0 \\ -1, \text{si} & x < 0 \end{cases} \qquad \text{Función de transferencia de tipo escalón}$$

La única neurona de salida del Perceptrón realiza la suma ponderada de las entradas, resta el umbral y pasa el resultado a una función de transferencia de tipo escalón. La regla de decisión es responder +1 si el patrón presentado pertenece a la clase A, o -1 si el patrón pertenece a la clase B. La salida dependerá de la entrada neta (suma de las entradas x_i ponderadas) y del valor del umbral θ.

Una técnica utilizada para analizar el comportamiento de redes como el perceptrón es representar en un mapa las regiones de decisión creadas en el espacio multidimensional de entradas a la red. En estas regiones se visualiza qué patrones pertenecen a una clase y cuáles a otra. El perceptrón separa las regiones por un hiperplano cuya ecuación queda determinada por los pesos de las conexiones y del valor umbral de la función de activación de la neurona. En este caso, los valores de los pesos pueden fijarse o adaptarse utilizando diferentes algoritmos de entrenamiento de la red.

Sin embargo, el Perceptrón, al constar sólo de una capa de entrada y otra de salida con una única neurona, tiene una capacidad de representación bastante limitada. Este modelo sólo es capaz de discriminar patrones muy sencillos.

Como ejemplo de funcionamiento de una red neuronal de tipo Perceptrón, veamos cómo resolver el problema de la función OR. Para esta función, la red debe ser capaz de devolver, a partir de los cuatro patrones de entrada, a qué clase pertenece cada uno. Es decir, para el patrón de entrada 00 debe devolver la clase 0 y para los restantes la clase 1. Para este caso, las entradas serán dos valores binarios. La salida que produce, sin tener en cuenta el valor umbral, es

$$y = f\left(\sum_i w_i x_i\right) = f\left(w_1 x_1 + w_2 x_2\right)$$

donde:

x_1, x_2 : son las entradas a la neurona(en las neuronas de la capa de entrada, la salida es igual a la entrada)

w_1, w_2 : son los pesos entre las neuronas de la capa de entrada y de la capa de salida

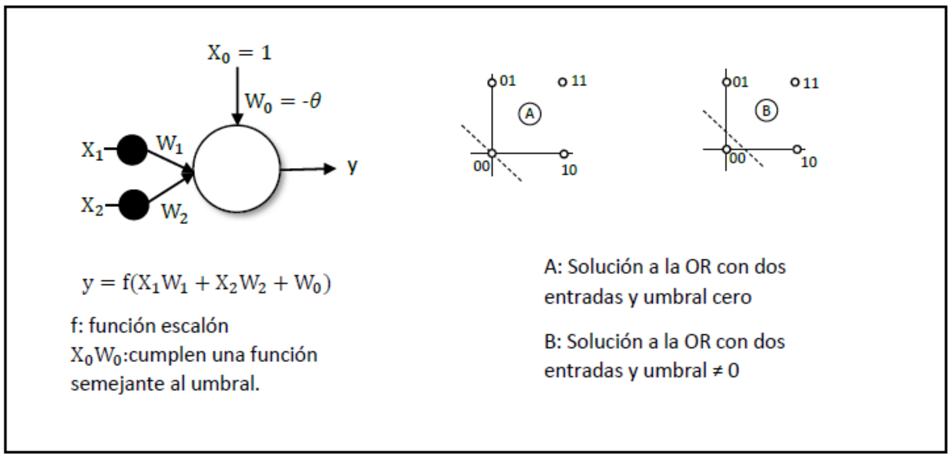
f : función de salida o de transferencia

Si $w_1x_1 + w_2x_2$ es mayor o igual a 0, la salida será 1, y en caso contrario, será -1 (función de salida en escalón). Como puede observarse, el sumatorio que se le pasa como parámetro (entrada total) a la función f (función de salida o transferencia) es la expresión matemática de una recta, donde w_1 y w_2 son variables y x_1 y x_2 son las constantes. En la etapa de aprendizaje se irán variando los valores de los pesos, obteniendo distintas rectas.

Lo que se pretende al modificar los pesos de las conexiones es encontrar una recta que divida el plano en dos espacios de las dos clases de valores de entrada. Concretamente, para la función **OR** se deben separar los valores 01, 10, 11 del valor 00. En este caso, al no existir término independiente en la ecuación porque el umbral θ es cero, las posibles rectas pasarán por el origen de coordenadas, por lo que la entrada 00 quedará sobre la propia recta.

Si se pretende resolver el problema de la función **AND** de la misma manera, se llega a la conclusión de que es imposible si el umbral es cero, ya que no existe ninguna recta que pase por el origen de coordenadas y que separe a los valores 00, 01 y 10 de entrada del valor 11, por lo que es necesario introducir un término independiente para poder realizar esta tarea.

Para ello, se considera una entrada de valor fijo 1 a través de una conexión con peso w_0, que representa el umbral ($w_0 = -\theta$) y cuyo valor deberá ser ajustado durante la etapa de aprendizaje. Así, el parámetro que se le pasa a la función de transferencia de la neurona queda:



Ejemplo del perceptrón aplicado a la solución de la función OR

$w_1x_1 + w_2x_2 + w_0(1)$, donde w_0 es el término independiente que permitirá desplazar la recta del origen de coordenadas. Si aplicamos esta solución para el caso de la red que calcula la función OR, aumentamos el número de soluciones, ya que, además de las rectas sin término independiente ($w_0 = 0$) que dan solución al problema, existirán otras con término independiente que también lo harán.

En el proceso de entrenamiento, el Perceptrón se expone a un conjunto de patrones de entrada, y los pesos de la red son ajustados de forma que, al final del entrenamiento, se obtengan las salidas esperadas para cada uno de esos patrones de entrada.

Una red neuronal es un campo escalar

$$f: \quad \mathbb{R}^n \to \mathbb{R}$$