

# Capítulo 1: Neural Network Acivation

## Resumen del capitulo

En el “*capitulo 1: Nueral Network Activation”* repasamos la forma en que se debe realizar el cálculo para determinar si una neurona es activada o no. Para empezar, explica que el operador se sumatoria funciona con un ciclo *for*, e indica que todos los términos contenidos se suman; de esta manera el cálculo de la activación de una neurona se realiza de manera similar, solo que se suman los productos de las entradas por los pesos correspondientes. Este resultado es pasado a la función de activación.

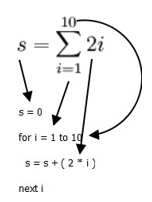
La función de activación procesa el dato proporcionado y proporciona una salida, esta salida corresponde al valor de activación. Las funciones de activación realizan un papel critico dentro de las redes neuronales, dado que le dan la capacidad, podríamos llamarlos de esta forma, de aprender, gracias a que los pesos que están funciones retornan son muy variados y oscilan naturalmente en los rangos -1 a 1 en la función tangente hiperbólica y de 0 a 1 en la función sigmoide. Esta capacidad es debido a la curva de que dibujan dichas funciones.

Por último, repasamos la utilización de las neuronas de sesgo, las cuales, permiten que la red genere valores de 0, y cabe rescatar que el valor que estas reciben siempre es 1.

Aclarando, todo lo anterior mencionado es refiriéndose a una red neuronal feedforward.

## Preguntas de revisión

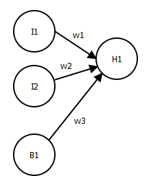
1. **¿Qué es el operador de sumatoria? ¿Cómo es utilizado?**

Representado por la letra sigma del alfabeto griego, es utilizado para representar una sucesión de sumas continas dado un valor inicial hasta uno final, respetando la regla representada en dicha expresión ejem:

Podemos afirmar que es utilizada de manera similar a un bucle for de programación.

1. **Ejemplifique como se realiza el cálculo de la activación de una neurona usando el operador de sumatoria.**

Tomando el ejemplo del libro; tenemos un fragmento de la red neuronal original:

**Donde:**

L1, L2, B1 corresponden a valores de entrada.

w1, w2, w3 son lo pesos correspondiente.

H1 neurona en donde las neuaronas L1,L2 y B1 hacen sinapsis

**Para calcular tenemos que:**



De tal manera que el calculo lo podemos hacer con el bucle for y quedaría de la siguiente manera:

double w[3]; //pesos

double i[3]; //entradas

double sum; //resultado

for (int c = 0; c < 3; c++){

sum += (w[c] \* i[c]);

}

1. **¿Qué es una función de activación?**

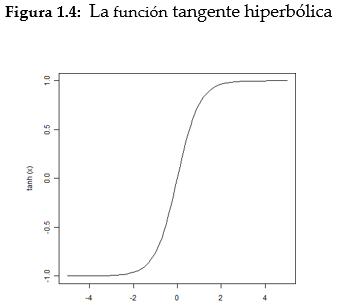
Función que se aplica después de realizar el calculo de una neurona, retorna un valor representante del valor de activación de dicha neurona. Estas funciones integran la no linealidad dentro de las redes neuronales permitiéndoles que estas tengan la capacidad de aprender.

1. **Describa las funciones de activación que se describen en el capitulo**

Para las redes neuronales feedforward tenemos dos:

**Tangente hiperbólica**

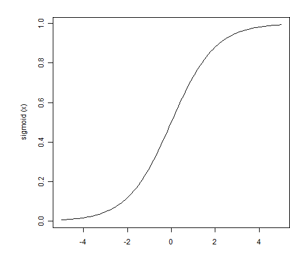
Es la mas común de usar, sus rangos van desde -1 a 1. Su grafico luce así:





**Sigmoide**

Sus valores van desde los rangos 0 a 1 se utiliza cuando es necesario trabajar con números positivos. La grafica luce así:





1. **¿Qué es una neurona de sesgo?**

Los valores de sesgo permiten que una red neuronal genere un valor de cero incluso cuando la entrada está cerca de uno. La adición de un sesgo permite que la salida de la función de activación se desplace a la izquierda o a la derecha en el eje x.

## Vocabulario

**Summation Operator**

Operador que representa la sumatoria consecutiva dando una expresión desde un valor inicial hasta uno final. Generalmente representado con la letra sigma del alfabeto griego, se asemeja mucho al bucle for de programación.

**Activation Functions**

Dado una entrada realizan un calculo y retornan un valor que representa el valor da activación de una neurona. Estas funciones permiten que las redes neuronales puedan aprender.

**Hyperbolic Tangent function**

Función que utiliza la formula de la tangente hiperbólica para el cálculo, sus rangos van del -1 a 1 y es una de las más usadas.



**Sigmoid function**

Función que es denominada función logística, utilizada en cuando se requiere trabajar con valores positivos, dado que su rango de valores esta entre 0 y 1.



**Bias Neurons**

Los valores de sesgo permiten que una red neuronal genere un valor de cero incluso cuando la entrada está cerca de uno. La adición de un sesgo permite que la salida de la función de activación se desplace a la izquierda o a la derecha en el eje x.

# Capítulo 6: Weight Initialization

## Resumen del capitulo

En la inicialización de los pesos de una red neuronal se debe tener en cuenta que estos valores no deben ser constantes, más bien aleatorios; el rango de -1 a 1 es muy popular. Para esto es muy popular el uso del algoritmo Ngunyen-Windrow, técnica que ajusta los pesos iniciales de la red. Si graficamos dichos pesos obtenemos un histograma, en el cual se nos da información sobre los pesos asignados a la red neuronal.

Ngunyen-Windrow funciona asignado pesos iniciales adaptativos, permitiendo que la red neuronal aprenda de manera más eficiente. Aunque este método es muy efectivo, no es el único que existe, además de no ser el mejor. Todo depende de que función realizara la red neuronal,

## Preguntas de revisión

1. **Explique ¿Por qué es importante la inicialización de pesos de una red neuronal?**

Inicializar los pesos en 0 de una red neuronal, no es la mejor opción que existe debido a que nunca van a mejor los pesos en el ajuste resultado que la red no tendrá la capacidad de aprender y no se alcanzará el objetivo planteado. Sin embargo, cuando los pesos se establecen de manera aleatoria la red tendrá un punto de partida y los ajustes realizados serán aceptables.

1. **¿Cuál es el algoritmo de inicialización de peos más popular? ¿Por qué?**

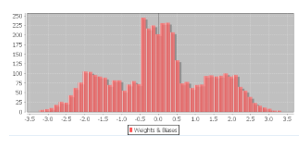
El algoritmo de inicialización de pesos mas popular es el algoritmo Nguyen-Windrow, dado que ajustas los pesos de manera mas adecuado asegurando menos iteraciones durante el entrenamiento de la red.

1. **¿Qué es un histograma? Dentro de redes neuronales ¿Qué uso le podemos dar?**

Grafico que nos ayuda a ver de manera gráfica la asignación de pesos de una red neuronal (hablando en materia de redes neuronales), aunque su aplicación no se restringe solo a este campo. Su función radica que en el eje Y muestra el numero de apariciones de un grupo determinado.

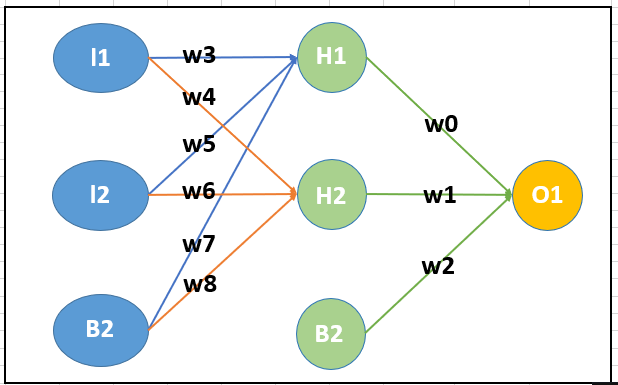
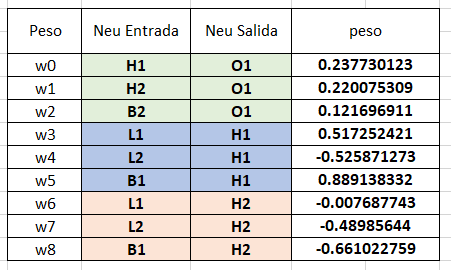
1. **¿Qué es el algortmo Ngunyen-Windrow?**

Algoritmo de inicialización de pesos que consiste en la asignación de pesos entre los rangos de -0.5 y 0.5, para después modificar dichos pesos. En esta modificación se podría observar un grafico semejante al de la figura.

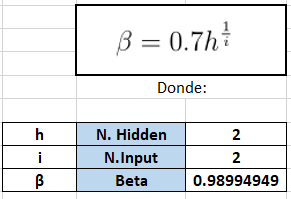


1. **Explique de manera breve como funciona el algoritmo de Ngunyen-Windrow.**

Antes que nada los valores de la matriz deben inicializarse en valores dentro del rango -0.5 a +0.5.

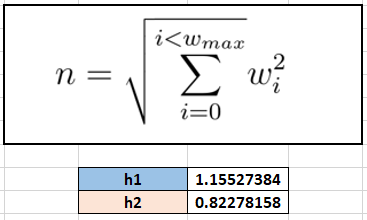
 

Ahora realizaremos el cálculo de beta

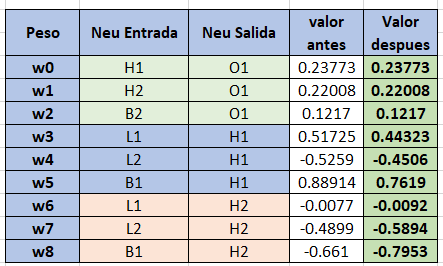


= 0.98994949

Y para cada neurona oculta calcularemos la norma euclidiana:



Para final mente realizar el reajuste de pesos de la matriz con se muestra en la siguiente imagen



= **0.443231515**

= **-0.450616974**

= **0.761899053**

= **-0.009249693**

= **-0.589382588**

= **-0.795325473**

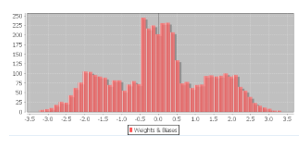
## Vocabulario

**Weight Initialization**

Asignación de peso a cada neurona de la red, generalmente esta asignación se hace mediante numero aleatorios, aunque es más optimo la utilización de algoritmos, como el algoritmo de inicialización Nguyen-Widrow.

**Histogram**

Grafico que representa la parición de un grupo dentro de un conjunto con respecto al eje y.



**Random numbers**

Números de índole aleatoria. En redes neuronales son utilizados para la asignación inicial de pesos, de tal manera que los valores no sigan patrones predispuestos, y así mejorar la eficacia del entrenamiento mediante algoritmos de inicialización da pesos.

**Ngunyen-Windrow**

Algoritmo de inicialización de pesos mas popular que se caracteriza por su optimo desempeño, reduciendo el numero de iteraciones que se deben realizar para llevar a cabo el entrenamiento de la red.

**Gaussian curve**

Se llama curva normal a la distribución gaussiana: la distribución de probabilidad de una variable continua que suele resultar próxima a un fenómeno real.